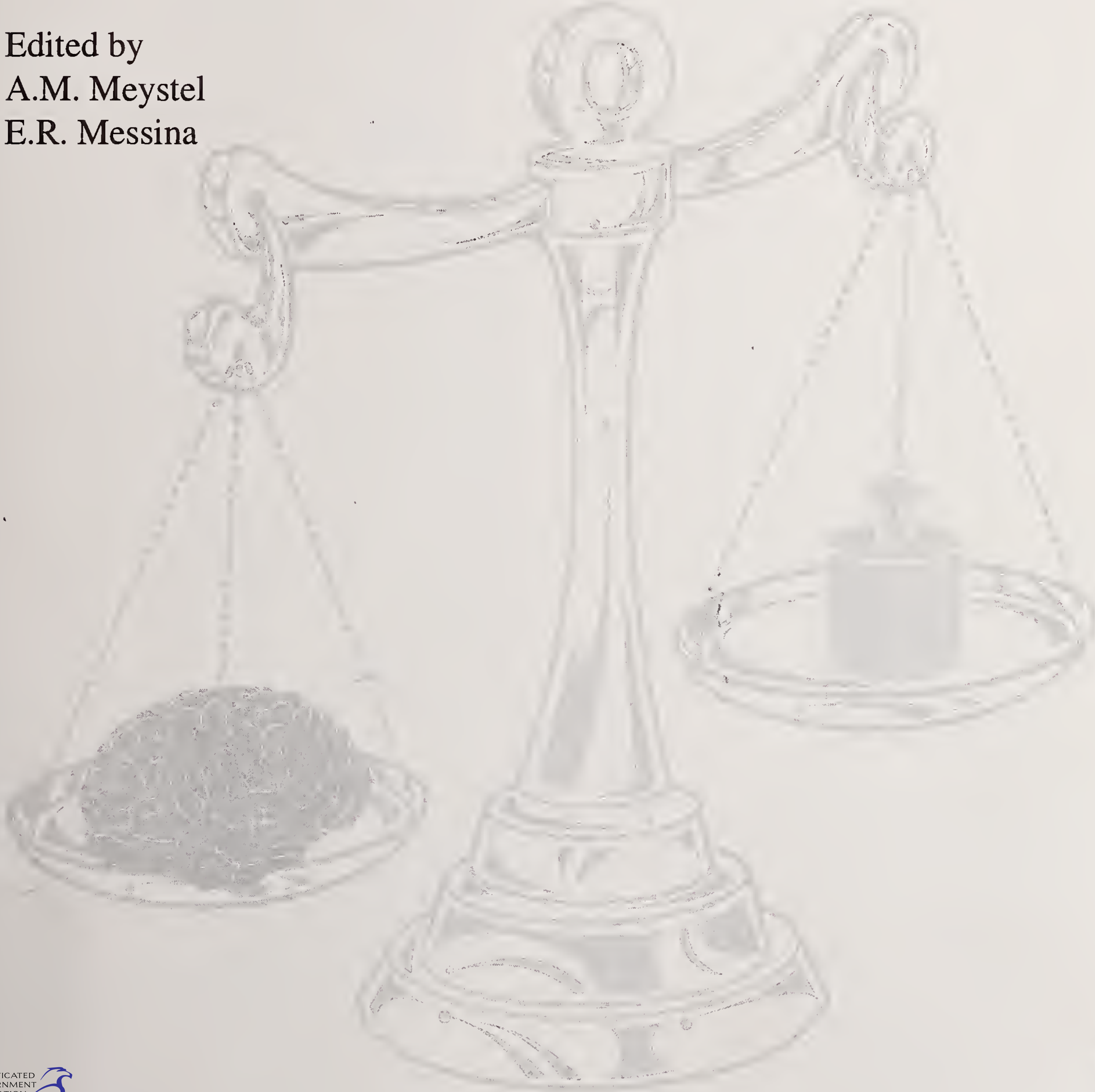
National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

NIST Special Publication 970

Measuring the Performance and Intelligence of Systems: Proceedings of the 2000 PerMIS Workshop August 14-16, 2000

Edited by
A.M. Meystel
E.R. Messina



The National Institute of Standards and Technology was established in 1988 by Congress to "assist industry in the development of technology . . . needed to improve product quality, to modernize manufacturing processes, to ensure product reliability . . . and to facilitate rapid commercialization . . . of products based on new scientific discoveries."

NIST, originally founded as the National Bureau of Standards in 1901, works to strengthen U.S. industry's competitiveness; advance science and engineering; and improve public health, safety, and the environment. One of the agency's basic functions is to develop, maintain, and retain custody of the national standards of measurement, and provide the means and methods for comparing standards used in science, engineering, manufacturing, commerce, industry, and education with the standards adopted or recognized by the Federal Government.

As an agency of the U.S. Commerce Department's Technology Administration, NIST conducts basic and applied research in the physical sciences and engineering, and develops measurement techniques, test methods, standards, and related services. The Institute does generic and precompetitive work on new and advanced technologies. NIST's research facilities are located at Gaithersburg, MD 20899, and at Boulder, CO 80303. Major technical operating units and their principal activities are listed below. For more information contact the Publications and Program Inquiries Desk, 301-975-3058.

Office of the Director
- National Quality Program
- International and Academic Affairs

Technology Services
- Standards Services
- Technology Partnerships
- Measurement Services
- Information Services

Advanced Technology Program
- Economic Assessment
- Information Technology and Applications
- Chemistry and Life Sciences
- Materials and Manufacturing Technology
- Electronics and Photonics Technology

Manufacturing Extension Partnership Program
- Regional Programs
- National Programs
- Program Development

Electronics and Electrical Engineering Laboratory
- Microelectronics
- Law Enforcement Standards
- Electricity
- Semiconductor Electronics
- Radio-Frequency Technology[1]
- Electromagnetic Technology[1]
- Optoelectronics[1]

Materials Science and Engineering Laboratory
- Intelligent Processing of Materials
- Ceramics
- Materials Reliability[1]
- Polymers
- Metallurgy
- NIST Center for Neutron Research

Chemical Science and Technology Laboratory
- Biotechnology
- Physical and Chemical Properties[2]
- Analytical Chemistry
- Process Measurements
- Surface and Microanalysis Science

Physics Laboratory
- Electron and Optical Physics
- Atomic Physics
- Optical Technology
- Ionizing Radiation
- Time and Frequency[1]
- Quantum Physics[1]

Manufacturing Engineering Laboratory
- Precision Engineering
- Manufacturing Metrology
- Intelligent Systems
- Fabrication Technology
- Manufacturing Systems Integration

Building and Fire Research Laboratory
- Applied Economics
- Structures
- Building Materials
- Building Environment
- Fire Safety Engineering
- Fire Science

Information Technology Laboratory
- Mathematical and Computational Sciences[2]
- Advanced Network Technologies
- Computer Security
- Information Access
- Convergent Information Systems
- Information Services and Computing
- Software Diagnostics and Conformance Testing
- Statistical Engineering

[1] At Boulder, CO 80303.

[2] Some elements at Boulder, CO.

NIST Special Publication 970

Measuring the Performance and Intelligence of Systems: Proceedings of the 2000 PerMIS Workshop August 14-16, 2000

Edited by:
A. M. Meystel
E. R. Messina
Intelligent Systems Division
Manufacturing Engineering Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8230

Co-Sponsored by:
National Institute of Standards and Technology
Defense Advanced Research Projects Agency
Institute of Electrical and Electronic Engineers Control Systems Society
National Aeronautics and Space Administration

In Cooperation with:
Institute of Electrical and Electronic Engineers Neural Network Council

Issued September 2001

U.S. Department of Commerce
Donald L. Evans, Secretary

National Institute of Standards and Technology
Karen H. Brown, Acting Director

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

National Institute of Standards and Technology Special Publication 970
Natl. Inst. Stand. Technol. Spec. Publ. 970, 648 pages (September 2001)
CODEN: NSPUE2

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 2001

For sale by the Superintendent of Documents, U.S. Government Printing Office
Internet: bookstore.gpo.gov — Phone: (202) 512-1800 — Fax: (202) 512-2250
Mail: Stop SSOP, Washington, DC 20402-0001

Preface

This volume contains the materials of the Performance Metrics for Intelligent Systems Workshop, held at the National Institute of Standards and Technology on August 14th through the 16th, 2000. The central theme of the meeting *was Measuring the Performance and Intelligence of Intelligent Systems*. The functioning of intelligent systems is driven by evaluation of the "success" of assigning and achieving the goals. Both the adequacy of *assigning* and the degree of *achieving* belong to the gray area of measuring performance. How well the system is designed for achieving its goals, and how effective and efficient the efforts are that its control system produces — these two issues belong to the domain of evaluating the degree of the intelligence of a system. Neither the system's performance, nor its intelligence can currently be adequately measured by evaluation techniques other than those generally used in control systems. Engineers and researchers are not satisfied with these approaches as they are applied to intelligent systems.

The Workshop was the first formal gathering of the professionals actively working and/or interested in this area. The problem is a multidisciplinary one in its essence. Therefore it should integrate both engineers and scientists actively working in diverse areas such as economics, artificial intelligence, psychology, linguistics, biology, neurology, and others. Unifying them for solving the problems of measuring performance and intelligence is a formidable problem: their interaction is the only avenue that can bring to fruition this area of the science of intelligent systems.

This volume starts with the White Paper (Part I) that initiated the process of communication among the multidisciplinary group of engineers and researchers. The papers, submitted and accepted for presentation are collected in Part II. Not all of them could be presented at the meeting because of the difficulties of traveling from all over the world. They are grouped corresponding to the sub-area of the problem. Notes made by the participants of the general Panel Discussions are collected in Part III. The decisions of the Advisory Board are presented in Part IV. Some results of the pre-workshop discussion are put together in the Appendix. We hope that this volume will help to continue the process of consolidating the efforts and precipitating the results of research and design in this innovative area of science. We will be grateful for the comments sent to us concerning the problem of measuring the performance and intelligence of intelligent systems.

We wish to acknowledge the support of the Defense Advanced Research Projects Agency (DARPA) Mobile Autonomous Robots Software Program. We are also very thankful to our partners and co-sponsors. The workshop was co-sponsored by the National Aeronautics and Space Administration, the Institute of Electrical and Electronic Engineers (IEEE), and DARPA, and organized in cooperation with the IEEE Neural Net Council. Our thanks go out to our Plenary Speakers: H. Szu, G. Saridis, J. Albus, S. Grossberg, and W. Freeman. We are grateful to all the participants and the very enthusiastic members of the Advisory Panel for their many and significant contributions. A great debt is owed to Debbie Russell for helping produce the proceedings and Aveline Allen for logistics support.

Editors:

A. Meystel and E. Messina October 23, 2000

TABLE OF CONTENTS

PART I

PART II

PART I

MEASURING PERFORMANCE AND INTELLIGENCE OF SYSTEMS WITH AUTONOMY

The White Paper

Measuring Performance and Intelligence of Systems with Autonomy: Metrics for Intelligence of Constructed Systems[1]

A White Paper Explaining Goals of the Workshop

1. Introduction

Thousands of person-years have been devoted to research and development in the various aspects of artificially intelligent systems. There is no single field of study that contributes to the progress, but rather several dozens, ranging from control to cognitive sciences. Much progress has been attained. However, there has been no means of evaluating the progress of the field. How can we assess the current state of the science? Some systems are beginning to be deployed commercially. How can a commercial buyer evaluate the advantages and disadvantages of the *intelligent* candidates and decide which system will perform best for their application? If constructing a system from existing components, how does one select the one that is most appropriate within the desired system?

The ability to measure the capabilities of intelligent systems or components is more than an exercise in satisfying intellectual or philosophical curiosity. Without measurements and subsequent quantitative evaluation, it is difficult to gauge progress.

It can even be argued that researchers and developers perpetually re-invent the same components to build their system, unable to reliably find existing components they could reuse. To paraphrase William, Lord Kelvin: when you can measure something and put some numbers to it, then you know something about it, and if you can't your understanding of it is of a "meager and unsatisfactory kind," although I am not sure that I would be so adamant about the need for numbers.

It is both in a spirit of scientific enquiry and for pragmatic motivations that we embark on the quest for metrics for intelligence of constructed systems.

[1] This paper is a result of collective efforts to understand the problem, and the future publication based on this paper will have multiple authors. The draft was written by A. Meystel. Initial editing was done by J. Albus, E. Messina, J. Evans, D. Fogel, and W. Hargrove. These are the authors of multiple additions to the initial draft: G. Bekey, H.-H. Bothe, B. Chandrasekaran, J. Cherniavsky, A. Clerentin, P. Davis, S.

2. Intelligent Systems (or Agents)

Intelligent systems (that are also frequently called "agents") can be introduced with different levels of detail. The simplest possible and the most general model of intelligence is just a string of six consecutively functioning elements forming a loop of closure: WORLD INTERFACE, SENSORS, PERCEPTION, WORLD MODEL, BEHAVIOR GENERATOR, and ACTUATOR. The loop of closure consisting of these six modules has a flow of *knowledge* circulating within this loop and changing its form within each of the modules. It is possible to demonstrate that if one introduces the concept of intelligent agent in this simple form, a significant degree of generality is achieved in talking about a single intelligent system as a part of the overall model of functioning. Let us try to define this loop with *knowledge* circulation in it, as a scientific entity. The subsequent description of an Intelligent Agent is relevant to our needs of analysis and design. This is the list of features characteristic for an intelligent agent.

Feature 1. Intelligence is the faculty of an agent that allows to deal with *knowledge* and to achieve the externally measurable *success* under a particular *goal*.

Feature 2. The knowledge of an agent is the collection and organization of information units. Knowledge is presumed to appear as a result of the *learning* about the objects of the external world, interconnections of the objects, and processes of changes produced by the agent within this external world. These processes are characteristic for all intelligent systems.

Feature 3. The learning process is understood as recording the *experiences* encountered by an intelligent system and deriving from these experiences a new set of rules that suggests how the intelligent system should act under particular circumstances (in a particular situation and under particular goal). **Feature 3A**. Learning provides for a successful adaptation of agent (intelligent system) to changing environments, e.g. different algorithms of new rules derivation can be utilized (i.e. algorithms of reinforcement, habituation, Hebbian association, abstraction, generalization, etc.).

Learning[2] invokes special metrics that affect the way of judging the performance and intelligence of systems with learning. In the machine learning community there is a tendency to look at three metrics: the ability to generalize, the performance level in the specific task being learned, and the speed of learning. From the point of view of evaluating intelligence, the ability to generalize seems to be the most important one. Systems can do rote learning, but without generalization, one cannot apply what has been learned to future situations. Of course, if two systems were equivalent in their ability to generalize, with the same resulting level of performance, then the one which could do this faster would be "better."

Feature 4. Experiences are understood and stored as triplets of the information units "situation→action→new situation" that allow the behavior generation module of the agent to infer what is the action that is required to improve the situation (evaluation is presumed).

Feature 5. A situation is understood and stored as a complete set of sensor inputs associated with a particular moment of time in a form that allows for processing. A situation also includes the entire situational

Delaroche, L. Erasmus, D. Filev, L. Fogel, W. Freeman, S. Grossberg, S. Lee, P. Lima, L. Pouchard, A. Schultz, C. Weisbin, A. Yavnai.

analysis, such as the operating goals, parameters, and hypotheses about external conditions, such as enemy locations.

Feature 6. All artifacts of learning are evaluated for their desirability according to the criteria of goodness existing in this particular agent.

Feature 7. Action of an agent results in a complete set of agent motion (or behaviors) that are developed by actuators of the agent and are sensed by the agent as changes in the external world.

Feature 8. The intelligent system (or the agent) is presumed to be equipped with the relevant sets of sensors and actuators, with the information storage, an inference system and a device for value judgment that allows for ranking both the experiences and the rules and determining their preference for the goal of the system.

One can see that no degree of sophistication is discussed in this setting. All processing is explained as inference, and various versions of inference will entail different levels of sophistication. One of the important mechanisms of inference is the mechanism of generalization: An agent is capable of inferring how to find an appropriate group of objects, how to transform it into a single object, and how to derive the rules for the generalized object from the rules that were known for its components.

So far, the described system looks very cozy and almost trivial in the very beginning of its existence. However, as the amount of experiences grows, the complexity of computations grows exponentially and the efficiency of goal-oriented functioning falls. No respectable agent would allow itself to be overburdened by growing complexity. This is why the operator of *generalization* is introduced: agents cannot afford the complexity of computations. This is the main reason for the emergence of mechanisms of generalization: they create new objects by the virtue of merging similar objects delivered and utilized by the original set of sensors and produced by actuators[3].

These generalized objects form a new world of representation: the one belonging to a lower level of resolution. As a result, we end up with a multiplicity of interrelated hierarchies of percepts, concepts, commands and actions. Corresponding multiscale systems of objects form a storage of the World Representation[4]. Any functioning actor has this system that provides its functioning.

Feature 9. The goal is the overall assignment to the system that determines the purpose of its functioning and the preferences that system uses to choose the action, and eventually determines the structure of its knowledge representation.

[2] Contributed by A. Schultz

[3] Generalization and abstraction occur on items resident in memory, in an indefinite amount of time. I reflect on events from last year, yesterday, and this morning, and may detect a pattern I hadn't noted before. This may be a higher-level generalization & abstraction than of the immediate kind applied to sensory inputs.

[4] We are familiar with the fact that some researchers disagree with the need for World Representation. It could be argued that all architectures are equipped by some form of World Representation, albeit under a different label.

Feature 10. In the system with a multiscale knowledge representation the action determined at the lower level of resolution becomes a goal for the higher level of resolution. Thus, we are used to the situation that the goal arrives from the exterior of each level of resolution.

Feature X. The unknown feature.

What this feature is follows from answering two questions that emerge immediately as soon as the first nine features are introduced:

Question X.1 Who creates the goal for the lowest level of resolution?

Question X.2 Can the goal be formulated internally (at a level of resolution)

The design of increasingly autonomous intelligent agents will also require an end-to-end approach, in which all the aspects of perception, cognition, emotion, and action are realized in a single system[5]. Feedback cycles of information processing need to be designed from perception through action and then back to perception again, mediated by feedback through the environment. Such cycles of information processing can evaluate the effects of system performance on the environment, and modify the system where needed to achieve better environmental control. It has also become clear that, in addition to these externally mediated cycles of information processing with the environment, internally mediated feedback is needed to achieve autonomous system properties. Such internal feedback realizes properties of intentionality and attention that are characteristic of biological intelligence.

Consciousness[6] might be considered as a possible candidate for interpreting the Feature X. This is one possible view on the contribution of consciousness as a feature (faculty) of intelligence. Only those creatures that adequately forecast their environment survive, that is, recognize the dangers and opportunities in time for a suitable reaction. Since the real world is dynamic and uncertain, having a feature for discovering new ways to solve new problems should be one of the key features of intelligence.

Consciousness provides a view of the *self* in the context of the immediate environment. As a capability it did not arise suddenly, but rather, establishes itself at different levels and in different degree. The dog understands his environment and his place within it with some degree of clarity. We know ourselves and our environment in more precise terms and can even include unseen elements. I'm conscious of the time of day, what happened yesterday, what might happen tomorrow, even what's happening in Serbia without having been there. It is consciousness that allows manipulations of alternative models of the real world as we understand them. Here is the basis for dealing with an enormous range of issues as they pertain to survival. The mechanism of consciousness seems to be the "software" of human intelligence.

The primary problem with respect to consciousness is the underlying algorithmic mechanism. This subject has received a lot of attention in recent years. The real challenge is to build a mechanism that is conscious, not simply simulates the behavior of a conscious entity. There is no homunculus within us. The question emerges, how does perception present itself to us as an integrated entity? How are we capable of understanding our own consciousness?

[5] These observations are taken from the abstract by S. Grossberg

[6] Contributed by L. Fogel

A related problem is concerned with "binding." In what manner are the various modalities (vision, hearing, and the other senses) combined when we now know that vision itself is compartmentalized with separate perception centers for color, shape, texture, and so forth. How can all this be done in real time? There are other intrinsic problems that are yet to be faced. An interesting question is, what will a higher level of consciousness be like, above and beyond what we now have? What if our species grows into something even more complex with greater intelligence? What would be the nature of self-awareness and understanding of the world in which it operates? Could a machine facilitate consciousness through some symbiotic relationship? There are more questions to be asked than answered. What are the links between survival and consciousness? Consciousness is essential in an n-player game wherein survival depends upon the induced behavior of other players and your relationship with them. Consciousness presumes a conscious ability. This too is an intrinsic aspect of intelligence and we expect that it shall be addressed.

3. The Problem of Measuring both Performance and Intelligence

Both engineered and organized - that is, artificially produced - intelligent systems should demonstrate qualities similar to those demonstrated by living creatures, and especially by humans: ability to work under a hierarchy of goals, and subgoals ability to perceive the external world and recognize objects, actions and situations, ability to reason, make decisions, plan, schedule and evaluate the results of actions and learn from their experiences. These systems are actually Constructed Systems with Autonomy (CSA); we will call them *Intelligent Systems*.

Intelligent Systems of interest have both their body and their mind designed by humans (engineers and programmers); we have to recognize which part of the intelligence is incorporated in their "body" and which is a faculty of their "mind" (i.e. its intelligent control system). The structure and the characteristics of the "body" can relax the requirements of the intelligent control system if the results of past experience of functioning or anticipated future situations are properly incorporated in the design. Proper distribution of systems' intelligence between body and mind is a part of engineering design. Different degrees of autonomy require different degrees of total intelligence, and a different distribution of total intelligence between the "body" and the "intelligent controller".

Intelligent Control Systems are usually equipped with a system of Perception (Sensory Processing), Knowledge Representation (where the world model is constructed, frequently in the form of the ontology), and Behavior Generation (that creates task decomposition, plans and issues commands). As a rule, these systems are multigranular (multiscale, multiresolutional), and they resolve their problems at various scales simultaneously.

Multiple existing definitions of intelligence emphasize different facets of this complex phenomenon. We will follow the definition of intelligence formulated by J. S. Albus in 1991: " intelligence will be defined as an ability of a system to act appropriately in an uncertain environment, where appropriate action is that which increases the probability of success, and success is the achievement of behavioral subgoals that support the system's ultimate goal."[7]

Intelligent Systems differ in the depth and the breadth of the "appropriateness" of acting they demonstrate in different situations. Subsequently, they differ in the degree of "success" they are capable of achieving. The functioning could be made more appropriate and the level of success could be improved if we understand how to measure their intelligence. Thus, the measure of intelligence can be frequently reduced to measuring the "success" of functioning as provided by the ability to develop "appropriate" activities of the constructed intelligent system. The problem is non-trivial as can be seen from the case study below. We intentionally have chosen an exotic example since most of the readers can construct much more sophisticated cases related to unmanned autonomous vehicles, cooperating multilink manipulators, space stations, robot-companions, etc.

The Albus definition of *intelligence* is based upon understanding of the term success[8]. The success of solving a given task depends on the system's faculties, plus on some influences, which might be of stochastic nature or might not be measurable. One group of faculties can be called "the capacity to solve problems" or *intellect.* Intelligence includes intellect and, in addition, a number of other faculties that together help to facilitate the *success.* These additional faculties of intelligence include a) sensing abilities, b) skills of sensory processing and image interpretation, c) the capacity to collect, store and organize knowledge, d) the ability to use knowledge, i.e. via problem solving and decision making processes; the latter includes developing of the alternatives of plans for future actions, evaluating their preferability and choosing one of them, e) the ability to transform the decision into actions that lead to a success. Thus, intelligence represents a 'potential ability' to solve a given task in good time. A high intellect might compensate for the lack or deficiencies in other components of intelligence, and vice versa.

Many concepts of measuring intelligence exist. Many were proposed in communications during preparation of this White Paper. This is what L. Fogel[9] suggested:

1) Intelligence is measured in terms of the diversity of purposes that can be achieved under the range of environments. This diversity is usually reflected in the number of dimensions in the Space of Intelligence (see Section 6). The greater the diversity of purposes/situational constraints, the greater the intelligence.
2) Measures of performance must be from the point of view of some social entity. Thus, the results of measuring the degree of success are very relative. Accomplishing a certain task (or range of tasks) may be of great value to Mr. A, and of little value to Mr. B. There can be no absolute metric.

[7] J. Albus, "Outline for a Theory of Intelligence," *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. 21, No. 3, May/June, 1991, pp. 473-509

[8] The subsequent consideration of the term *success* was proposed by H. -H. Bothe

[9] From an e-mail message, May 30, 2000

3) The worth of performance must include the cost of the performing. In some cases, this is merely operational cost, in others its R&D, T&E, acquisition and installation, as well as operations. Rarely this cost may include removal minus salvage value.

One aspect of this integrated mechanism of *intelligence* as commonly understood is that the agent who has it is often able to produce behavior that has a certain *reasonableness* to it[10]. That is, if one knew the goal of the agent, one might agree that the behavior was oriented to achieving the goal. A. Newell identified this quality of intelligence as a kind of *rationality*. He then asked what made the agent successful in achieving the goal. The answer was: the agent had knowledge and had some ways of using the knowledge for the goals.

The "way of using the knowledge" can be interpreted as and is embodied in the agent's *architecture*. He then noted that sometimes an agents's knowledge is bound up for use for only certain types of goals. On the other hand, for some agents, some of the knowledge is available for any goal for which it is potentially applicable. Chandrasekaran gives an example of a visual system that has knowledge that elements of the visual scene that have similar velocities probably belong to one object. However, while we "have" this knowledge in some sense, it is typically not available for us to reason with in our deliberative problem solving. It is simply hard-wired for use only for certain problems in vision[11].

On the other hand, we know many things explicitly. And as long as our memory doesn't fail us, we are often able to use our explicit knowledge for many different goals for which the knowledge is relevant. In the case of humans, we have a deliberative cognitive architecture that can often retrieve the relevant knowledge and make it available for the explicit (conscious) problem solving.

A. Newell proposed that an idealized version of an intelligence (in the sense of *rationality*) would always use knowledge K if it had it and if it was relevant for a goal. This is purely an architectural characterization: it doesn't say anything about what kinds of knowledge are useful. If an agent has a certain goal, if knowledge K is useful for it, and if it doesn't have it, the agent of course won't use the knowledge. But the agent probably has some other knowledge K', which may be used to generate a subgoal of identifying the knowledge needed and maybe acquiring it[12]. With the appropriate ways of interacting with the world, the agent would use knowledge K' first, and then acquire the knowledge K, and voila, the goal is achieved.

Focusing on the ability to use knowledge for any relevant goal characterized, for A. Newell, is an extremely important aspect of intelligence. We would like to notice that one more faculty of intelligence is involved: namely, focusing attention, which is frequently used by the agent in its search activities.

[10] This discussion of the interpretation of the term intelligence was contributed by B. Chandrasekaran

[11] While this thought is powerful and probably correct, the example is not particularly persuasive. It is hard to say whether this knowledge is utilized by the subject that visualizes the scene. One might assume that we group the adjacent points together into one object not because they have the same speed but, on the contrary, we deduce that they have the same speed because they belong to the same object. The grouping for declaring the fact of "belonging to the same object" might be done by the virtue of spatial adjacency no matter what the speeds of the points actually are.

[12] This formidable conjecture is based upon an assumption that the agent somehow *knows* that by achieving a subgoal, knowledge about how to achieve the main goal will be acquired. Given the current state of practice, it would be more natural to assume that a problem solving intelligence should be equipped by a faculty of searching, and in situations where knowledge is lacking, it develops a set of searching activities.

This sense of intelligence goes against a common intuition in which intelligence is associated with having the knowledge rather than the ability to use the knowledge you have to acquire the knowledge (i. e. to focus attention and search). Later, we will discover that when we focus our attention and get engaged in searching, usually we end up with finding groups of similarity and create clusters[13] — objects of the lower resolution.

Newell's definition deserves our attention because it captures one sense of the term in a way to which some sort of metric may be attached. Purely reactive machines -- which map their perceptions directly into actions, such as the thermostat -- are on the low end of the scale. Further up are machines that can map from perceptions to actions by considering a large but precompiled number of alternative paths that are constructed by grouping, while groups are found by search and focusing attention.

4. A Case Study: Artificial Climate System

In an Artificial Climate System, it is required to maintain the temperature of the air in the controlled rooms within some interval of temperature $\Theta°$ (with some accuracy $\Delta\Theta°$), and provide the value of humidity within some interval of h (with some accuracy Δh) for a particular moment of time t. In addition, the Artificial Climate System should keep some function within some interval $F_t(\Theta °, \Delta \Theta °, h, \Delta h, t) \leq \Delta F$ experimentally determined to be preferable for a human being. In this case, the goal pursued by the system is not a particular state $S_t(\Theta °, \Delta \Theta °, h, \Delta h, t)$ but is rather an unknown function $F_t(\Theta°, \Delta \Theta °, h, \Delta h, t) \leq \Delta F$.

This problem is rather a nontrivial one. It can be compared with a problem of welding control where the function of the seam quality is very complicated and typically unknown since it depends on many factors, some of them hard to measure, or even evaluate. Generally, the problem is similar to the problem of optimum control of all multivariable stochastic controllers with incomplete available information that do not pursue a particular state but rather being within an interval of some cost-function. The explicit or implicit ability of a system to generalize might be crucially important for providing a proper functioning of the system and maintain the proper climate to the full satisfaction of the user.

Even more complicated functioning can be expected if this cost function is unknown, and the system of Artificial Climate should learn it by observing the behavior of the human user. This would require observing how many times the human user was turning "on" or "off" the ventilator, how many times the user was turning "on" or "off" the cooling unit, the humidifier, and what were the measures of temperature and humidity at these moments in the room. A simplistic automated system might be confused, but an intelligent system with elements of learning will pursue a mutually satisfactory schedule of functioning for all interrelated subsystems. The system will in fact learn the climate related "personalities" of the users and will learn to recognize who demands what and when. Even more bold generalizations could be expected if the system can correlate the user's behavior with the readings of temperature and humidity outside (not only inside!).

The goal of this learning process should be reduction of the amount of human intervention — that is, increasing the autonomy of the system. If the human-user needs to tune the system less frequently, this would

[13] One of the elements of new knowledge generation.

mean that the system works better. An even more interesting situation might happen if there is more than one user, and different users have different policies of tuning the system up, i.e. multiple users have different propensities in intervening with the Artificial Climate System. The Artificial Climate System that would minimize the total number of cases of human intervention would be considered a system for achieving consensus in a particular multi-player game.

A further development of this system might be required if the owner of this particular hotel wants actually to reduce the cost of energy required for keeping the customers satisfied. Then, the system can be designed so that it will learn habits of the customers to keep their average number of complaints below some particular level, while the energy consumed will be minimized. We can see that all these systems have a pretty high degree of autonomy: they autonomously assign the schedule of subsystems functioning. On the other hand, these systems are *subserviently autonomous*, i. e. they control their own behavior but the goals are totally determined from the external user.

The solution of this problem might be different for the systems that have a sense of *self*. A system may be considered to have a sense of self if it is equipped to take into consideration its own interests or advantage — and generate goals and success criteria for itself. Initially, we consider a set of regular obedient controllers that are intelligent (to a degree) but do not have any *self*, yet. The system equipped with a *self*, will try to keep all sources of assignment satisfied (including customers and the hotel owner) while worrying primarily about enhancement of its own life span (reducing aging, increasing reliability, and so on). In other words, a further development of the system presumes its self-evolving and self-improving.

This Artificial Climate System with elements of autonomy can be qualified as an intelligent one. It definitely should have elements of learning, should have an ability to recognize phenomena of the external world that are required for its functioning, must use elements of deductive and inductive reasoning, and must generalize upon the input information and the results of its own functioning. We can see that the "intelligence of the system" can grow, as the goal of functioning grows in its dimensionality and levels of detail. We can judge the degree of intelligence by the breadth and depth of the goals that are achieved and the performance measures that are satisfied. We are not only interested in evaluating the correspondence between the goal and performance criteria on one side and the degree of intelligence on another side. We are interested in tools that allow for the growth of intelligence and more adequate satisfaction of the assignment.

5. The System Specifications and Vector of Performance (VP)

One specific property of intelligent systems is lack of knowledge about the future conditions of functioning. The list of variables is incomplete, the intervals of future parameter changes are uncertain, the goals to be pursued can be formulated only in general. Lack of clarity in design specifications calls for design redundancy which amounts to the need for autonomously compensating for uncertain control specifications and vaguely specified contingencies.

The system requirements identify the characteristics which the Intelligent System (e. g. unmanned ground vehicle) must possess. The choice of the specific components from the Tools of Intelligence (see the subsection on that topic) mandates which of the following capabilities are included to satisfy the specific system requirements:

- to recognize objects, actions, situations
- to infer from the recognized elements of the scene
- to search for a required object within a scene
- to remember scenes and experiences
- to interpret situations
- to evaluate objects, actions, situations, and experiences
- to learn new skills from positive and negative experiences
- to generalize upon recorded similarities and acquire new concepts
- to detect an unfamiliar object, label it, and then learn about it
- to communicate with humans and other intelligent systems
- to collaborate with humans and other intelligent systems
- to interpret its own behavior
- to adapt to new environment
- to interpret behavior of other intelligent systems
- to properly generate a solution in an unexpected situation
- to perform task decomposition
- to plan and schedule in time planned activities
- to support all modes of planning/control required.

Other system requirements can be deemed pertinent to the general architectures of intelligent systems. It seems practical to construct the Vector of Performance (VP) for each of the subsystems in full correspondence with the subsystem's specifications. We always know *quantitatively* what the outputs of interest are. The set of these outputs forms the target vector VP_T. Within the space of performance there corresponds to some particular area: the zone of performance determined by the set of specifications. After testing the real i-th system or systems we receive a real vector or set of vectors {Vi} that are supposed to be compared with VP_T.

The result of this comparison is the result of measuring a concrete V_i by determining the degree of its belonging to the zone of the performance space occupied by VP_T. Note that this is not a standard single-dimensional conventional measurement when a particular unit of measurement is introduced. Rather, this is determining the membership function in a class.

The mathematics of comparison does exist. It is not frequently applied to the realistic cases because it is not frequently requested by the professionals who are responsible for the evaluation and comparison of complex systems. However, for some particular subsystems the comparison between $\{V_i\}$ and VP_T is a common practice. We refer to the area of control systems where many comparison metrics have been developed. Some additional effort would be required to apply a similar approach for more general and difficult cases but this effort is within our reach.

In the area of intelligent systems, an additional difficulty is expected linked with the fact that a concrete system is always a hierarchy of subsystems. For each particular subsystem chosen within a concrete research and/or industrial domain, the comparison between $\{V_i\}$ and VP_T is well understood. However, not much thought was given yet to the mathematics of integrating V_i and VP_T of subsystems into V_i and VP_T of the overall system. We are optimistic about development of the appropriate techniques. In many real situations, this has been done in practice. It would be appropriate to expand the experience from real situations to the general theory of (hierarchical) vector comparison since real situations affect the architectural issues in a more relevant manner.

6. Intelligence, Goals Hierarchy, and Arbitration

A device with a very low level of "intelligence," can perform its duties and achieve the goals in an excellent way within the boundaries of its "obtuseness." Yet, a very intelligent device with the ability to make powerful generalizations of the available information, capable of performing a sophisticated processing of this information and generating new concepts often cannot perform the task as well as a simple "obtuse" device, for example, maintenance of the temperature in the room within a concrete interval. This very intelligent device starts interfering with the level of humidity, looks for correlation links between recent commands of the human operator, and doing other things that the user does not need. Thus the user response: what is the merit of "intelligence" if the job has not been done or has not been performed in a timely manner (i. e. within the specified concrete interval)? Similar things happen with humans when an overeducated person is assigned for a simplistic job.

Intelligent behavior is characterized by flexible and creative pursuit of endogenously defined goals[14]. It has emerged in humans through the stages of evolution that are manifested in the brains and behaviors of other animals. Intentionality is a key concept by which to link brain dynamics to goal-directed behavior. The archetypal form of intentional behavior is an act of observation through time and space, by which information is sought for the guidance of future action. Sequences of such acts constitute the key desired property of free-roving, semi-autonomous devices capable of exploring remote environments that are inhospitable for humans. Intentionality consists of the neurodynamics by which images are created of future states as goals, of command sequences by which to act in pursuit of goals, of predicted changes in sensory input resulting from intended actions (reafference) by which to evaluate performance, and modification of the device by itself for learning from the consequences of its intended actions. Imagination images, i. e. the images of the future states produced by the planner and/or the predictor, or the results of simulation can be produced in the form in which the SP system would see if the actions were carried out, or in a symbolic form of topographical map representation (at the lower resolution), or even in a descriptive form (at the lowest level of resolution).

Intelligent Systems are to be used in cases that are too complicated for using simple controllers; otherwise simple programmed and/or automated devices should be used. A notion of *closed* vs. *open* systems should be introduced that is relevant to the situations where programmed vs. intelligent devices can be utilized. *Closed systems* can be characterized by having a clear assignment of the problem to be solved, and a crisp

[14] From the abstract submitted by W. Freeman

ability to be characterized by a complete list of concrete user specifications in the terms of measurable variables. These are the cases where using an intelligent system is excessive.

On the contrary, in an *open system*:

- the problem is not totally clear
- its parts are not concretized; decomposition is not obvious
- the variables are not listed in the beginning of design process
- many variables will emerge during the process of functioning
- the methods of their observations and registration are not known *a priori*
- many rules of action should be learned during the process of functioning.

So far, we can indicate two diametrically opposite strategies exercised by intelligent systems: one strategy is characterized by a very long-term general goal, say, survival of a system, another by a set of short term particular goals. The strategy of survival demands that intelligent systems be able to adapt to the environment and all circumstances. The strategy of "adapting no matter what" determines particular laws of an intelligent systems's functioning. The other strategy is "following particular goals" no matter what. The latter strategy frequently leads to the destruction of the system at hand: it might perish while following its goals persistently. Adaptation is not possible under the second alternative of intelligent systems since adaptation demands a compromise of the particular short term goals that the system was assigned.

There is an intuitive feeling that the systems with the second strategy are somehow better, or preferable than the systems that adapt no matter what. However, this intuitive feeling is difficult to rationalize and explain. Obviously, these goals belong to different levels of granularity (scale, resolution) and they can be reconciled only by considering the larger scope of the situation. Following the particular goal no matter what may lead to the destruction of this particular system but will provide for survival of the rest of the team of intelligent systems (e.g. a squad of unmanned autonomous vehicles; in other examples analogous situation takes place, i.e. as the problem gets complicated, the solution moves to the domain of multiagent solutions).

Therefore, these two strategies can be compared with respect to some additional criterion that has a higher priority than "just survival", or "just pursuing the goal." One of such external criteria is that of "knowledge acquisition." Under this criterion, one should carefully analyze the very intention to survive while abandoning the goal, or an intention to achieve the goal, even if the perspective of being destroyed is actually an imminent one. Both intentions might turn out to be secondary issues if the rate of knowledge acquisition is at stake, and in one case this rate was higher than in another. Indeed, one can adapt to the details of surrounding environment even without knowing the broader world.

In the meantime, while the system is studying the world and ardently acquires knowledge of it, the model of the world evolves so much that a *simple* adaptation is merely impossible[15], and the survival is achieved for the system that has *evolved.* Negotiation is a powerful tool that allows for adjusting the intentions (toward the goal achievement) to the rational evaluation of the losses that might occur if the goals are pursued persistently and incessantly.

The possibility of negotiation and arbitration generates more complex scenarios[16]. Assume an agent "wins" a particular negotiation at a given time. It would also allow for (but not require) that the tie-breaking arbitration assures that all of the goals are brought to the attention of negotiating parties over time. The arbitrator might want to make sure that a given agent wins "something," especially after losing out several times. Or, in goal terms, the arbitrator might be concerned about maintaining a balance "in portfolio terms". If there were goals associated with efficiency and discovery, the arbitrator might keep track of how cumulative efficiency and discovery supplement the awards of goals achievement. If, as the result of a number of decisions over time, efficiency was always winning out, and the locker of discovery items was empty, then the arbitrator could adjust his tie-breaking rules. This means that autonomous intelligent agents should not always try to be (locally) efficient, especially if they are equipped with learning.

On the other hand, we may be getting intelligences and goals of our agents mixed up. Suppose that there is a set of goals (G_1,...G_n). Different agents might have different pure goals, or they might put different weights on the various goals. Further, they might be better or poorer at pursuing those goals in differing contexts. That is, they might have different components of intelligence (I_1,I_2,..I_s) and these would be more or less important in the different contexts (C_1,...C_q). Indeed, a human may value beauty, order, material things, family, and learning new things, just to mention a few items.

This human might be very good at aesthetic matters and family matters, but not so good at order and material things. The agent might be good at trying and learning about new aesthetics-related things, but poor at doing anything risky. It is typical for humans to have a portfolio of "intelligences" as well as "goals." It would give some value to all the different goals, and would have some value to each dimension of intelligence. Another human might be characterized as an explorer, although he would value family and wealth to some degree--just not as much as new discoveries. Yet a third might be an explorer in search of tidiness (e.g. a scientist). What do you think, which human will do better? It depends. An unequivocal answer might be impossible at a single level of resolution because the true result depends on the distribution of the types of agents and the contexts that the groups of agents find themselves in.

7. What Constitutes the Vector of Intelligence (VI)?

We are still in limbo about what we should measure to evaluate intelligence: the mysterious Vector of Intelligence (VI), or the system's success asattributable to its intelligence. (The need to construct a VI emerges in many areas.)

For example, the problem of the appropriate degrees of generalization, granularity, and gradations of intelligence occurs in ontology development[17]. What constitutes the appropriate scope and levels of detail in an ontology is practically driven by the purpose of the ontology. The ability to dynamically assume one level of

[15] *Adaptation* is understood as a mere parametric adjustment while the evolutionary changes in the *structure* of a system are results of *learning.*

[16] Contributed by P. Davis

[17] Submitted by L. Pouchard

detail among many possible details is important for an intelligent system. It might depend on the purpose of a system. In that sense the long term purpose of the system is different from its short term or middle term goals. Clearly, the long term purpose and the multiple term goals are goals belonging to different levels of resolution and should be treated in this way. This brings us back to the measures of intelligence through success: is intelligence to be measured by the ability of a system to succeed in carrying out its goals?

The term "success" is a key word in the Albus definition, because it becomes a source of emerging gradations in intelligence, the degrees of intelligence depend on the essence of the definition of the word *success*. This means that if success is defined as producing a summary of the situation (a generalized representation of it), the summary can be computed in a very non-intelligent manner especially if one is dealing with a relatively simple situation. Indeed, in primitive cases, the user might be satisfied by composing a summary defined as a "list of the objects and relationships among them" i.e. a subset of an entity-relational (ER) network[18]. On the other hand, the summary can be produced intelligently by generalizing the list of objects and relationships to the required degree of quantitative compression with the required level[19] of the context related *coherence*. Thus, *success* characterizes *intelligence* if the notion of *success* is clearly defined.

The need in gradations of intelligence is obvious: we must understand why the probability of success increases, because somebody is supposed to provide for this increase, and somebody is supposed to pay for it. This is the primary goal of our effort in developing the metrics for intelligence. The problem is that we do not yet know the basis for these gradations and are not too active in fighting this ignorance. What are these gradations, how should they be organized, what are their parameters that should be taken into account? We can introduce parameters such that each of the parameters affects the process of problem solving and serves to characterize the faculty of intelligence at the same time.

The following list of 25 items should be considered an example of the set of coordinates for a possible Vector of Intelligence:

(a) memory temporal depth

(b) number of objects that can be stored (number of information units that can be handled)

(c) number of levels of granularity in the system of representation

(d) the vicinity of associative links taken into account during reasoning of a situation, or

(e) the density of associative links that can be measured by the average number of Entity-Relation (ER)-links related to a particular object, or

(f) the vicinity of the object in which the linkages are assigned and stored (associative depth)

(g) the diameter of associations ball (circle)

[18] See the summaries produced by the search-engines on the Web: to have it "quick and dirty" the first sentence, or the first 5 lines of an article is considered to be a summary, why not?

[19] Summarizing an article (in unstructured natural language), if done properly, is a result of generalizing the natural text description and transforming a narrative from one level of resolution by a narrative from another level of resolution.

The association depth does not necessarily work positively, to the advantage of the system. It can be detrimental for the system because if the number of associative links is excessively large the speed of problem solving can be substantially reduced. Thus, a new parameter can be introduced

(h) the ability to assign the optimum depth of associations

(This is one more example of recognition that should be performed, in this case, within the knowledge representation system).

Functioning of the behavior generation module evokes additional parameters, properties and features:

(i) the horizon of planning at each level of resolution

(j) the horizon of extrapolation at a level of resolution

(k) the response time

(This factor should not be confused with a horizon of prediction, or forecasting which should combine both planning and extrapolation of recognized tendencies).

(l) the size of the spatial scope of attention

(This corresponds to the vicinity of the associative links pertinent to the situation in the system of knowledge representation)

The following parameters of interest can be tentatively listed for the sensory processing module:

(m) the depth of details taken into account during the processes of recognition at a single level of resolution

(n) the number of levels of resolution that should be taken into account during the processes of recognition

(o) the ratio between the scales of adjacent and consecutive levels of resolution

(p) the size of the scope in the most rough scale and the minimum distinguishable unit in the most accurate (highest resolution) scale

It might happen that recognition at a single level of resolution is more efficient computationally than if several levels of resolution are involved. A finer system of *inner* multiple levels of resolution can be introduced at a particular level of resolution assigned for the overall system (e.g. Burt's pyramids[20]). The latter case is similar to the case of unnecessarily increasing the number of associative links during the organization of knowledge.

Spatio-temporal horizons in knowledge organization as well as behavior generation are supposed to be linked with spatio-temporal scopes admitted for running algorithms of generalization (e.g. clustering). Indeed, we do not cluster the whole world but only the subset of it which falls within our scope. This joint dependence of clustering on both spatial relations and the expectation of their temporal existence can lead to non-trivial results.

One should not forget that generalization (the ability to come up with a "gestalt" concept) is conducted by recognizing an object within the chaos of available spatio-temporal information, or a more general object within the multiplicity of less general ones. The system has to recognize such a representative object, event, or action if they are entities. If the scope of attention is too small, the system might not be able to recognize the entity that has boundaries beyond the scope of attention. However, if the scope is excessively large, then the system will perform a substantial and unnecessary job (of searching and tentatively grouping units of information with weak links to the units of importance).

Thus, any system should choose the value of the horizon of generalization (that is the scope of the procedure of *focusing of attention*) at each level of resolution (granularity, or scale).

All of these parameters characterize the realities of the world and the mechanisms of modeling that we apply to this world. These parameters do not affect the user's specifications of the problem to be solved in this system. The problem is usually formulated in the terms of hereditary modeling that might not coincide with the optimum modeling, or with the parameters of modeling accepted in the standard toolbox of a decision-maker.

The problem formulated by a user often presumes a particular history of the evolution of variables available for the needs of the intelligent system. Simultaneously, the user requests a particular spatio-temporal zone within which the solution of the problem is desirable. However, the input specifications often do not require a particular decomposition of the system into resolution levels and the intelligent system is free to select it in an "optimal" way. In other cases, the user comes up with an already existing decomposition of the system that appeared historically and must not be changed (like the organizational hierarchy of a company and/or an Army unit). Sometimes, it is beneficial to combine both existing realistic resolution levels and the "optimal" resolution levels implied by the optimum problem solving processes.

The discrepancy between these decompositions requires a new parameter of intelligence

(q) an ability of problem solving intelligence to adjust its multi-scale organization to the hereditary hierarchy of the system, this property can be called "a flexibility of intelligence"; this property characterizes the ability of the system focus its resources around proper domains of information.

In the list of specifications of the problem the important parameters are

(r) dimensionality of the problem (the number of variables to be taken into account)

(s) accuracy of the variables

(t) coherence of the representation constructed upon these variables

For the part of the problem related to maintenance of the symbolic system, it is important to watch the

(u) limit on the quantity of texts available for the problem solver for extracting description of the system[21]

and this is equally applicable for the cases where the problem is supposed to be solved either by a system developer, or by the intelligent system during its functioning.

[20] P. J. Burt, "Multiresolution Techniques for Image Representation, Analysis, and 'Smart' Transmission," SPIE Conference 1199: Visual Communications and Image Processing IV, Philadelphia, Nov. 1989.

[21] Most of the input knowledge arrives in the form of stories about the situation. These stories are organized as a narrative and can be considered *texts*. In engineering practice, the significance of the narrative is frequently (traditionally) discarded. Problem solvers use knowledge that has been already extracted from the text. How? Typically, this issue is never addressed. Now, the existing tools of text

(v) frequency of sampling and the dimensionality of the vector of sampling

Finally, the user might have its vision of the cost-functions of his interest. This vision can be different from the vision of the problem solver. Usually, the problem solver will add to the user's cost-function of the system an additional cost-function that would characterize the time and/or complexity of computations, and eventually the cost of solving the problem. Thus, additional parameters:

(w) cost-functions (cost-functionals)

(x) constraints upon all parameters

(y) cost-function of solving the problem

This contains many structural measures. We need to trace back from an externally perceived measure of "success" or intelligence to a structural requirement. E.g, the construction codes specify thickness of structural members, but these dimensions are related to the amount of weight to support — the performance goal is the lack of building collapse.

Important properties of the Intelligent Systems are their ability to learn from the available information about the system to be analyzed. This ability is determined by the ability to recognize regularities and irregularities within the available information. Both regularities and irregularities are transformed afterwards into the new units of information. The spatio-temporal horizons of Intelligent Systems turn out to be critical for these processes of recognition and learning.

Metrics for intelligence are expected to integrate all of these parameters of intelligence in a comprehensive and quantitatively applicable form. Now, the set $\{VI_{ij}\}$ would allow us even to require a particular target vector of intelligence $\{VI_T\}$ and find the mapping $\{VI_T\} \rightarrow \{VI_{ij}\}$ and eventually, to raise an issue of design: how to construct an intelligent machine that will provide for a minimum cost (C) mapping

$$[\{VP_T\} \rightarrow \{VI_{ij}\}] \rightarrow \min C.$$

By the way, has this ever been done for the systems that are genuinely intelligent? Of course, this question is not related to design, just to measurement.

8. The Tools of Computational Intelligence

Proper testing procedures should be associated with the model of intelligence presumed in the particular case of intelligence evaluation. It seems to be meaningful to compare systems of intelligence that are equipped with similar tools. In this section we introduce the list of the tools that are known from the common industrial and research practice of running the systems with elements of autonomy and intelligence. It is also expected that these tools can be used as components of the intelligent systems architectures. Thus, they might help in developing and applying types of architectures that will be used for comparing intelligence of systems.

The following tools are known from the literature as proven theoretical and practical carriers of the properties of intelligence:

- Using Automata as a Generalized Model for Analysis, Design, and Control

processing allow us to address this issue systematically and with a help of the computer tools of text processing.

- Applying Multiresolutional (Multiscale, Multigranular) Approach
    1. Resolution, Scale, Granularity: Methods of Interval Mathematics
    2. Grouping: Classification, Clustering, Aggregation
    3. Focusing of Attention
    4. Combinatorial Search
    5. Generalization
    6. Instantiation
- Reducing Computational Complexity
- Dealing with Uncertainty by
    - Implanted compensation at a level (feedback controller)
    - Using Nested Fuzzy Models with multiscale error representation
- Equipping the System with Knowledge Representation
- Learning and Reasoning Upon Representation
- Using bio-neuro-morphic methodologies
- General Properties of Reasoning

  Quantitative as well as qualitative reasoning

  Generation of limited suggestions, as well as temporal reasoning

  Construction both direct and indirect chaining tautologies (inferences)

  Employing non-monotonic as well as monotonic reasoning

  Inferencing both from direct experiences as well as by analogy, and

  Utilizing both certain as well as plausible reasoning in the form of

    1. Qualitative Reasoning
    2. Theorem Proving
    3. Temporal Reasoning
    4. Nonmonotonic Reasoning
    5. Probabilistic Inference
    6. Possibilistic Inference
    7. Analogical Inference
    8. Plausible Reasoning: Abduction, Evidential Reasoning
    9. Neural, Fuzzy, and Neuro-Fuzzy Inferences
    10. Embedded Functions of an Agent: Comparison and Selection

Each of the tools mentioned in the list allows for a number of comprehensive embodiments by using standard or advanced software and hardware modules. Thus a possibility of constructing a language of architectural modules can be considered for future efforts in this direction.

9. The Architectures of Intelligence

Listings of all tools of computational intelligence presently available and all properties of intelligence measurable would not characterize the system exhaustively and would not suggest how to test the system. How these tools are attached to each other — this is what matters! It turns out that the architecture of the system can be decisive in providing active features of various intelligent systems.

Architectures of intelligent systems should support:

- Expected long-term mission planning (e.g. overall path planning and replanning for the whole mission performance)
- Various principles of knowledge representation
- Navigation, guidance and motion control with self-orientation using a set of techniques specified by the mission
- Auxiliary activities which require using additional intelligent control systems (e.g. for manipulator arms installed at the mobile autonomous platform)
- Ability to acquire the data, which characterize and quantitatively measure mission performance
- Perception capabilities: the character of the architecture will be strongly affected by the characteristics of all the sensors to be installed on-board of the autonomous intelligent system (for example, the unmanned ground vehicle); its intelligence will be affected by the designer's decision regarding what particular vision and other off-the-shelf perception systems are to be implemented, what is the level of human supervision[22] expected in the system (full autonomy, partial teleoperation, full teleoperation, etc.)
- Ability to handle sensing, data-processing, and decision making (including planning, navigation, guidance, and control), dealing with uncertainties, especially while operating in the uncertain environment
- Ability to respond to changes in the environment or its self-state without requiring human intervention.
- Ability to optimize performance based upon some cost-function (e.g. minimum time of task execution, minimum energy consumption, minimum final error of performance, minimum risk of being detected and/or destroyed[23])
- Multi-robot (multi-vehicle, multi-system) coordination
- Robot-supervisor interaction (in a multi-robot case this may entail robots-supervisor interaction, robots-supervisors interaction[24], etc.)

[22] A human supervisor will directly or indirectly assist the function of perception of the first group of unmanned ground vehicles.

[23] Often, all five of these factors are important: in this case weights must be assigned. However, some theoretical difficulties should be overcome before using this case in practice.

[24] In addition to the question: how should the interaction proceed among the members of the robotic team. One can ask a similar question about the team of human operators supervising the robotic team.

- Ability to perform a variety of tasks (e.g. in the case of unmanned vehicles, the ability to perform travel, reconnaissance operations, mine neutralizing, etc.)
- Fault-tolerant, reliable, and robust operation
- Measurable architecture performance both qualitatively and quantitatively[25]
- Extensibility for improvements and adaptation to mission specifics

Other information processing functions will probably need to be supported but those listed above most strongly affect the choice of architectural approaches. It is especially relevant in the cases where we are explicitly talking about dealing with knowledge.

The first group of these implicit architectural matters[26] includes *principles of knowledge representation* accepted in a particular intelligent system. A case could be made for semantic-based knowledge representation, including tests for completeness and consistency. Although the theories for such tests exist (e.g. Process Specification Language (PSL's) completeness and consistency can be proved within situation calculus). The breadth and scope of knowledge represented in a knowledge representation system also determines and conditions its possible re-use. Perhaps re-usable devices and software processes should be considered, since such processes potentially decrease costs of further systems. One might expect that re-usability criteria could be required for characterizing the intelligence.

Another group focuses more explicitly on ontologies that demonstrate the results of generalization within the stored linguistic information. Ontology development aims at building a machine-readable semantic layer within a (software) system. Ontologies formally express the knowledge contained in an application by providing definitions for concepts, relations and functions, as well as rules for constraining the use of the terms. Ontologies contain definitions for metadata and rules that constrain the interpretation and use of metadata. Ontologies can represent relations of inheritance, aggregation and instantiation.

Ontology development supports system interoperability by solving problems related to semantic ambiguities, and by enabling semantic communication between software agents. Software agents may refer to a common ontology to exchange messages. Actually, ontologies do not carry anything different in principle from all hierarchical constructions within the knowledge base. However, they present it in a language form, for some ontologies even in a natural language form. This opens an opportunity to communicate with large and "interdisciplinary" knowledge bases in natural language.

Providing translation mechanisms for the interoperability of applications requires that applications share a common ontology or that application concepts can be represented in a formal, declarative manner. Other benefits of ontologies include reliable system specifications, accurate data and metadata descriptions, and

[25] This requirement should not be confused with the functional requirement of measurability of performing a particular function, and/or the overall mission such as time of arrival, or fuel consumed, or percentage of mines neutralized. Here we are talking about performance of the architecture that should be measured in terms of performing intelligent control operations (e.g. computations per alternative of solution, goodness of solutions found, etc.).

[26] Submitted by L. Pouchard

development of common data formats for collaborative analysis. Ontologies that exist for specific tasks or domains permit knowledge sharing and re-use within the domain.

The scales and scalability criteria critical for intelligent systems are represented within ontologies, too.

10. Supervisory Control and Data Acquisition

Supervisory Control and Data Acquisition involves data collection, active communication with the user, and display. This is a group of separate subsystems (actually, several levels of the architecture) within the intelligent controller. These subsystems can be equipped by additional control loops and a separate knowledge organization system required for communication. The purposes of these subsystems are:

- to prepare information relevant to the needs of corresponding levels of control and command
- to convey this information to the user or the supervisory controller
- to conduct the dialog with the corresponding level of control and command
- to display all the information in a user friendly form e.g. use of graphics, use of previously negotiated modes of demonstration and protocol of explaining the ongoing activities
- to provide alarming, warning, notification both to other subsystems as well as for the external levels of control and command
- to provide for security by allowing different levels of control and command with different privileges.
- to facilitate printing and reporting functions, storage and display of historical data to facilitate investigation of events, investigation, and other types of analysis.

11. Tests of Machine Intelligence Contemplated in the Past

1. The Turing test, or *imitation game* was proposed by A.Turing in 1950[27]. In one version of this test a human judge interrogates a program through an interface. If the program can fool the human into believing that responses come from another human and not from a computer then the program should be considered intelligent. Clearly, in this test we don't talk about intelligence as a phenomenon but rather about an ability of pretending to be intelligent. At the present time, such an approach seems to be a naive one: it determines what *seems* to be intelligent rather than what *is* intelligent.

Nevertheless, this approach has generated a lot of literature, in particular the famous problem of Chinese room[28]. J. Searle considers the following mental experiment. A person was given a set of formal rules for manipulating Chinese hieroglyphs. This person does not speak or understand written Chinese, and he does not know the meaning of these hieroglyphs, he just can distinguish them visually[29]. The rules state that if a symbol of a certain shape is given to him, he should write down another particular hieroglyph on a piece of

[27] A. Turing, "Computing Machinery and Intelligence", *Mind*, Vol. 59, No. 236, October, 1950, pp. 433-460

[28] J. Searle, (1980) "Minds, Brains, and Programs", *Behavioral and Brain Sciences*,

paper. The rules prescribe how the groups of hieroglyphs should correspond one to another. When a set of Chinese symbols enters from outside, the person applies the rules, writes down a set of other Chinese symbols as specified by the rules, and returns the result to the external observer. The external observer perceives the result as a grammatically correct answer in Chinese. However, the person inside does not understand Chinese. (Note that the very possibility of conducting this experiment in reality is questionable: the list of required rules would be prohibitively large if the scope of questions and required answers covers a broad domain and demands for a high degree of sophistication).

Searle believes that the person in the Chinese room does exactly what a computer would be doing if it used the same rules to engage in a grammatically correct conversation in Chinese. Both the computer and our "inside" person are engaging in "mindless" symbol manipulation. This mental experiment leads J. Searle to the following statements:

Axiom 1: Computer programs are formal (syntactic) and manipulate *symbols.*

Axiom 2: Human minds have mental contents (semantics) and manipulate *meanings.*

Axiom 3: Syntax is not translated into semantics, therefore symbol manipulation does not contain any *understanding.*

Searle's argument is intended to show that implementing a computational algorithm that is *formally* isomorphic to human thought processes cannot be sufficient to reproduce the real process of *thought.* The last decade of research in the area of intelligent systems demonstrated that this reasoning is too simplistic and is not sufficient to adequately represent even existing constructed systems with autonomy (like unmanned autonomous vehicles). Searle's schemes of analyzing processes of"thinking" are overly primitive and cannot represent existing mechanisms of sensory processing, knowledge representation and behavior generation in multiresolutional systems of motion control practiced in existing autonomous vehicles. Something more is required. Researchers that develop intelligent systems challenge Searle's argument by creating new artifacts.

2. L. Zadeh's test can be formulated as follows: a paper is presented to the intelligent system, and it is supposed to transform it into a summary[30]. The quality of the summary can be judged by the ability of the system to generalize and formulate the meaning of the paper in a sufficiently concise form. No doubt, any system that can do it should be considered intelligent. Clearly, the system should be capable of generalizing. Says L. Zadeh: " the ability to manipulate fuzzy sets and the consequent summarizing capability constitutes one of the most important assets of the human mind as well as the fundamental characteristic that distinguishes human intelligence from the type of machine intelligence that is embodied in present-day digital computers[31]."

3. Various tests can be proposed based upon more mundane but more practical evaluations of sophistication and rationality. For example, we can check a capability of a program to generate several alternative decisions for a particular situation, and to select one of them properly; or its capabilities to analyze

[29] A subtle detail: distinguishing and recognizing most of the hieroglyphs is a serious intellectual problem by itself!

[30] L. A. Zadeh, from his BISC letter of 1999

[31] L. A. Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes," *IEEE Trans. on Systems, Man and Cybernetics,* Vol. SMC-3, 1973, pp. 28-44

the experimental data related to a particular physical system, and to compute a feedforward control, and to introduce a law of feedback compensation. The key issue in the last case is the ability to use the experimental data: different experimental data require different approaches to computing feedforward control, and different laws of feedback compensation. The tradeoff "feedforward vs. feedback" is a real test of intelligence as a tool for reaching successful balance under conditions of redundancy and uncertainty.

4. A. Newell has listed properties that intelligent system must have[32]:

- recognize and make sense of a scene
- understand a sentence
- construct a correct response from the perceived situation
- form a sentence that is both comprehensible and carrying a meaning of the selected response
- represent a situation internally
- be able to do tasks that require discovering relevant knowledge.

12. Who wins the competition: the Real Intelligence or the Impostor?

Using the Turing Test to evaluate intelligence has become commonplace, although as we have already mentioned above, it does not evaluate intelligence but rather the ability of a system to *pretend being intelligent.* Competitions are one of the straightforward primitive methods of judging the degree of intelligence. The deficiencies of competition are clear from the following list:

- in a competition, a random set of circumstances can affect the results rather than a set of capabilities of the competing systems; thus, only the results of multiple competitions can be valid
- it is difficult, if not impossible, to separate the part of intelligence endowed in the body design from the part of intelligence incorporated into the system of intelligent control; thus, for judging the intelligent control system, identical bodies are presumed
- competition in the natural environment cannot guarantee the equality of the problems to be encountered by competing parties; in constructed (artificial) environments, the difficulty of the problem drops drastically; it does not require that much "intelligence"

The latter feature is not necessarily always the case. The actual challenge is to provide a rich enough environment within which the tests can be conducted. An example of this would be a completely instrumented test course for evaluating autonomous robot mobility and mapping abilities, rather than the simple "box world" that is frequently used. In fact, one of the keys to our efforts in performance metrics is to come up with these sufficiently rich environments (test courses or very detailed, ground-truth simulation environments) which can be used to evaluate the performance of different systems. It is not an easy task. We should encourage a broad discussion on defining requirements for such environments.

[32] Newell, A. (1982). The knowledge level. *Artificial Intelligence*. 18(1), 87-127; Newell, A. and Simon, H. (1963), GPS: A program that simulates human thought, In *Computers and Thought* , ed. Feigenbaum and Feldman. McGraw-Hill, New York.

Therefore, winning a competition, however exciting it might be, leads to the old pitfall of the Turing test: winning requires no more than pretending to be intelligent rather than demonstrating real tools of intelligence. Testing of intelligence is a must, but the way of testing is a matter of discussion. The challenge for competitions is to overcome these obstacles. For example, developing an artificial environment that is dynamic and challenging , yet reproducible.

13. Measuring the Intelligence Contemplated for the Future

Measuring Intellifactors. One can start analyzing the problem of measuring intelligence within the domain of Albus's definition that assigns this faculty for control purposes[33]. The factors of intelligence are the factors of processes that contribute to intelligence (intellifactors). Logistically, they are dimensions of VI, mathematically, we can express this as follows:

$\mathbf{X_i} = \{\mathbf{x} | \mathbf{x}$ is all possible intellifactors$\}$

and the set of intellifactors $\{\mathbf{X_{if}}\}$, is an element of the power set of $\mathbf{X_i}$.

A measure of intelligence (IQ) is the measure that can assign a real number to the collective performance of each element in the set X_i. The measure of intellifactor (IFQ) is a measure that assigns a real value to the collective performance of each element in X_{if}.

Measuring the Power of Generalization. There exists a way to narrow the gap between building an intelligent machine (with its ontogeny[34]) and understanding the intelligence process by itself (with its epistemology[35]). The way is to model the process in a biological system[36]. How do brains do that? Brains avoid catastrophic failure when the complexity of computations grows exponentially by use of the NN-dynamics for generalization by creating "objects" (classes). It is experimentally confirmed that for the same operation of generalization, computer elements need more computations than brain needs. One can judge on the comparative productivity of computers during simple maps generalization[37] and instantaneous gestalt insights performed by the brain during human processing of complex images.

Measuring the System's Intelligence by the Degree of Uncertainty. The latter observation is linked with the entropy based considerations. Any measure of uncertainty (entropy in particular) is an acceptable measure of intelligence. If one can measure our uncertainty in taking decisions among alternatives, one can reduce this value of uncertainty (e.g., by learning), so our system is intelligent. But how do we measure the value of each alternative? Again, by its uncertainty. A possible way is to measure the probability of success of

[33] This concept of measuring intelligence was contributed by Louwrence Erasmus.

[34] or how it is done in a living organism

[35] or how it is done in the theory of knowledge

[36] This concept was proposed by W. Freeman. He refers to the A. Meystel's statement "the mechanism of generalization to emerge: it creates new objects" quoted from his e-mail letters to Advisory Board Members.

[37] J. D. McMahill, Interactive Generalization: User's Guide, CMU, Pittsburgh, PA 1998; G. L. Bundy, C. B. Jones, E. Furse, "Holistic Generalization of Large Scale Cartographic Data," in J. Muller, J. Lagrange, R. Weibel (eds.), *GIS and Generalization Methodology and Practice,* Taylor and Francis, London, 1995, pp. 106-119

meeting the specifications for each of the alternatives by successive applications or using a model (we might call it Reliability, in this sense). The higher the success, the lower the uncertainty/entropy. We may counter-balance this with the cost (or complexity) of achieving very successful alternatives (typically, the higher the reliability, the higher the cost)[38].

Constructing the Benchmarks. Judgment of the system's intelligence can be done by using indirect, albeit easy to measure values. In constructing benchmarks, we use the fact that the fundamental attributes of intelligence include:

- Ability to perform tasks in unstructured environments
- Ability to learn from experience
- Ability to transfer knowledge from one domain to another
- Ability to solve complex problems, requiring deductive and inductive reasoning

The following simple measures can be used as metrics for such abilities in machines[39]:

- Size and complexity of programs required
- Memory requirement
- Solution time

Clearly, such measures are useful only if (a) they are applied to benchmark problems, (b) all contestants use the same type and model of computer, and (c) all programs are written by comparably competent programmers, so that the programs are optimal in some sense.

Given these constraints, we could test intelligent systems A and B on the same benchmarks. The one that accomplishes the task more quickly, and does so with the least complex programs and least memory will be declared "more intelligent". While evaluating the level of intelligence based on this definition (to avoid the confusion of introducing a new one) we have to take into account[40]:

- type of uncertain environment
- strategy of achieving the goals
- capability of the system to automatically create and update its subgoals.

Most of the well-established methods for robust control design provide the capability to deal with small parametric and structural uncertainties and therefore represent a basic level of intelligence in the control system according to the definition of Albus. Situational uncertainty, e.g. drastic changes in the environment that are due to completely different operating conditions, severe and unpredictable disturbances, etc., completely alter system dynamics, and therefore require control systems with a much higher level of intelligence.

[38] The latter considerations were suggested by P. A. Lima

[39] Contributed by G. Bekey

[40] From the abstract submitted by D. Filev

Measuring Autonomy vs. Intelligence. The following question can be considered a fundamental one[41]: What is more important and meaningful to define and to measure with respect to the context of Intelligent Autonomous Constructed System— Autonomy or Intelligence of a Constructed System? We are looking for Autonomy, as the premier requirement of an Intelligent Autonomous System. From the designer or the user point of view, Intelligence enables Autonomy, but it is not a system design objective or a system requirement *per se.*

The definition of Autonomy is probably more precisely measurable and more meaningful and it is easier to come to a consensus about what Autonomy or an Autonomous System is all about, rather than what is Intelligence or an Intelligent System.

14. Simulated Functioning and Scaled Hardware Testing of Intelligent Systems

The hope is for a balanced combination of a) thorough simulation and b) scaled hardware testing. Many researchers focus upon simulating systems with high autonomy[42], like B. Zeigler in USA, K.-H. Brassel in Germany, I. Peters in Switzerland, J.-H. Kim and T.-G. Kim in Korea, and others. However, the challenge of evaluating intelligence of these systems remains an active problem to be resolved in the upcoming decade.

The most intricate problems associated with the variability and combinatorics of realistic situations can be resolved by simulating these situations. Thus even the predicament of absent hardware can be avoided by simulating the problem-impregnated situations. Contests and competitions can be considered a part of this paradigm. One cannot come even anywhere near covering in realistic testing the spectrum of philosophical[43] views of intelligence (just start to read the mind/body literature!) On the other hand, one might be inclined to scale back the possible analogies to human intelligence and human involved testing to less convenient but more pragmatic scenarios.

The Paradigm of Contest and Competitions

1. Symbolic systems. The a-y classification of measurable characteristics (see Section 7) can be made very representative but is definitely too constrained by the existing general systems and ways of representing information. Indeed, each of the 25 items on this list is a strong reduction of actual possibilities. Start with (a) memory temporal depth: why it should be limited? or why should only one value of depth be considered? The next item is (b) number of objects that can be stored: why should this number be limited? Then, we come to the number of levels of granularity, definitely a limitation that should depend on the problem. Then, we face limitation on the vicinity of associative links — the latter should not be limited as well! All 25 items on this list limit the opportunity to find better (not to speak about "the best") solutions. In the meantime, the environment

[41] From A. Yavnai's abstract

[42] See in Ed. by H. Sarjoughian, F. Cellier, M. Marefat, J. Rozenblit., *2000 AI, Simulation and Planning in High Autonomy Systems*, Proc. of the SCS Conference in Tucson, AZ, March, 2000

[43] From the e-mail letters by J. Cherniavsky

conducive for contests and competitions is by definition oriented toward a permanent atmosphere of inventiveness and development of new signs and new phenomena to be encoded by these signs.

2. Systems with learning. Learning is never forgotten as a very important subsystem of intelligence. However, there is not too much discussion related to the nature of learning as a substitute for real contests and competitions. In the meantime, learning plays the role of rehearsing expected ("would be") situations of contest and competition. Learning via prior experiences or via planning is a mechanism that prepares a system for contingencies. Thus, learning serves as a critical characteristic of intelligence that solely determines both the success and failure. It's there, but it's primarily implicit and serves as a supportive system that serves rather for improving functioning. Learning provides for a successful adaptation of the intelligent system to changing environments, e.g. different algorithms for deriving new rules can be utilized for different cases (i.e. algorithms of reinforcement, habituation, Hebbian association, abstraction, generalization, etc.). A multiplicity of situations can be anticipated where, without learning, the central purpose of the system could not be achieved.

3. Application Focused Intelligence. In many cases, the intelligence might be defined relative to a domain of application. Even in the human cases there are people who are "car intelligent" but "literature ignorant" - different domains, different abilities. This generates a question: if in the human domain one might distinguish different types of intelligence (Gardner's 7, Sternberg's 3, etc.) — should it be beneficial to try something similar in the autonomous unmanned, or partially manned systems? Indeed, for a human, the need to quickly move from one subject-oriented vocabulary to another might create a need to deal with using domain oriented algorithms of generalization, or pattern recognition. Can it be beneficial in the unmanned cases?

All three of these questions can be resolved within the domain of contests and competitions. We can create and focus on a specific domain where things like self-sustained, appropriate behavior, ability to quickly act in an uncertain environment, etc. can be physically quantified by realistic measures of performance (units of time, money, energy). The various contests (AAAI urban search and rescue, robotic soccer, the data-mining contests, the information retrieval competitions, the speech understanding rallies, etc.) provide the plausible level to measure and thus compare systems.

15. The Intelligence of Sensing and Sensory Processing

Available results have already suggested that the brain designs for sensory and cognitive processes differ from, and are even computationally complementary to, the designs for spatial navigation and action. This complementarity can be noticed by observing that cognitive knowledge needs to accumulate in a stable way over a period of years, with new knowledge not accidentally erasing previously learned, but still useful, knowledge[44].

The problem of data fusion (both heterogeneous or homogeneous) generated a demand that the robustness of the fusion stage be closely linked to the number of significant criteria permitting to associate information required for interpretation[45]. Both the uncertainty and the error of the input data, as well as

[44] From the abstract by S. Grossberg

[45] Contributed by A. Clerentin and L. Delahoche

uncertainties and errors of the available internal knowledge, jointly produce the uncertainty and the error of interpretation. The uncertainty is meant to characterize the "degree of actual existence" of the data; the error characterizes imprecision on the numerical evaluation of the data. The uncertainty and error estimation in classical fusion processes are generally based on a probabilistic approach. As the number of factors to be associated for interpretation grows, the need to work with multi-criteria techniques grows. The latter should help to evaluate the performance of each stage of global fusion processing: for example, data fusion for localization (generally allows for heterogeneous fusion) or data fusion for incremental map building (generally demands for homogeneous fusion: the same kind of primitives must merge on different acquisitions). Here again the use of tools like Dempster-Shafer theory of evidence might be promising.

In a number of applications, including the area of autonomous robotics, the problem of multi-sensor fusion and joint interpretation determines the value of intelligence related to sensing and sensory processing. It is clear that, in many situations, the use of multiple sensors is the only way of dealing with the richness of the external world. Any given sensor takes information about only one of the many attributes of the environment. But often the arriving information must be carefully gleaned for more than one attribute simultaneously. Only in this case can the required depth of interpretation be achieved.

So, the problem is how to integrate the information, especially when the sensors are disparate and when the viewpoints and even scales of incoming information are different. To overcome these problems, several fusion methods are used. The majority use a probabilistic approach (Bayes rules). A significant portion use a possibilistic approach that considers sensor evidence to be the value of belief (these rely on Dempster-Shafer theory). This theory is appropriately expressive, it explicitly represents ignorance, enabling the robot to differentiate between ambiguous sensing results and not having sensed at all. Other approaches include fuzzy logic or neural networks.

Information fusion is a growing research domain and of the numerous developed applications show that it enhances the level of autonomy and intelligence of engineered systems, especially autonomous robots.

16. Questions To Be Answered

This is the list of questions that the Workshop will try to answer[46]:

Question 1. What is the vector of intelligence (VI) that should be measured and possibly used as a metric for systems comparison?

Question 2. Should VI be measured in addition to, or instead of, measuring the vector of performance (VP) determined by the standard specifications?

Question 3. If two systems have the same VP, what is implied by the difference in their VI values? Can this difference be represented in monetary (cost) units?

Question 4. Is it possible (and meaningful) to have different VI measures: a) goal-invariant, b) resource-invariant, c) time-invariant?

[46] Questions 4, 6, 7, 8 were contributed by S. Lee

Question 5. What should be recommended as a test of VI and how can VP be normalized so that comparisons may be performed at the same normalized value of VP?

Question 6. Does a universal measure of system intelligence exist such that the intelligence of a system can be compared independently of the given goals[47]? A goal-independent measure may be more difficult to define. A goal-dependent measure, however abstract the goal may be, can allow for a clear comparison among the systems of different architecture but with the same goal. For instance, for the latter case, an intelligence can be represented as how efficiently, and how optimally a system reaches the given goal by itself, i.e., the power of automatically solving problems defined as the discrepancy between the goal and the current state.

Question 7. Should the intelligence measure of a system be solely based on problem-solving capability at time "t" or should it contain the potential increase of problem-solving capability in the future based on learning?

Question 8. Should the resources required for building systems and system operation play a role in defining the measure of intelligence? As mentioned above, the efficiency in problem solving should be included in the measure: for instance, the time and energy required to reach a solution should be taken into consideration together with the optimality of the solution. But, it is not clear whether we should or should not include the cost of building a system.

As a reminder, a set of other questions that are ingrained (directly, or indirectly) in the main questions is formulated as follows:

Question 9. These are the less profound ("secondary") questions that should be addressed at the workshop and possibly unequivocally answered:

a) how to form VI for various architectures?

b) should the questions 1 through 5 be related to intelligent systems, or autonomous systems, or both?

c) what is the protocol for dealing with uncertainty when the uncertainty metric is to be applied in the procedures of decision making? for example, how does the uncertainty of planning affect the cost of goal achievement?

d) what are the guidelines in constructing the world model and determining its scope in the variety of applications? how does the scope of "world model" affect the sophistication of intelligent behavior?

e) how are the questions 1 through 5 related to (and the answers applied to) the systems that are working under a hierarchy of goals?

f) should a competition between intelligent systems be considered a valid method of judging VI value?

[47] This seems to be hard to achieve for biological systems. This will be eventually addressed, but in the short term run the concrete goal of particular cases seems to be more attainable. A single measure of intelligence requires constructing a system of meta-knowledge.

17. Glossary

Autonomy — an ability to generate one's own purposes without any instruction from outside *(L. Fogel)*

Alternative definitions:

a) independence.

b) Self-government or the right of self-government; self determination.

c) Self-government with respect to local or internal affairs (AHD) ;

d) the right of self-government,

e) self-directing freedom (Merriam-Webster)

Autonomous System — a constructed system is autonomous if there is a likelihood that circumstances will arise in which no-one can predict in advance what it will do. *(T. Whalen)*

Autonomous Intelligent System - an autonomous constructed system is intelligent if we can be reasonably confident that whatever unpredictable thing it does do will be something that tends toward success in the goals for which the system was constructed in the first place. *(T. Whalen)*

Agent (sometimes **Autonomous, Intelligent**) — a term that has been introduced to use the word system which is regarded by many as a less desirable one when the software is involved, especially the one with properties of intelligence. The term Agent has some anthropomorphic overtones, Agent is presumed to be a system that probably can sense, reason and is intended to act. In other words, Agent should be understood as a system with elements of intelligence and autonomy.

Intelligence - an ability of a system to act appropriately in an uncertain environment, where appropriate action is that which increases the probability of success, and success is the achievement of behavioral subgoals that support the system's ultimate goal *(J. Albus)*

Alternative definitions:

- the ability to solve new problems in new ways *(L. Fogel)*
- the capacity to acquire and apply knowledge (AHD).
- the faculty of thought and reason (AHD).
- the ability to adapt effectively to the environment, either by making a change in oneself or by changing the environment or finding a new one (Britannica).
- the ability to learn or understand or to deal with new or trying situations (MWD)
- the skilled use of reason (MWD)
- the ability to apply knowledge to manipulate one's environment or to think abstractly as measured by objective criteria (MWD)

18. Appendix

How is Testing of Intelligence Performed on Humans?

The most widely used intelligence tests include the Stanford-Binet (SB) Intelligence Scale and the Wechsler Scales (WS). The Stanford-Binet test was first introduced in 1916 by Lewis Terman from Stanford University. The individually administered test, revised in 1937, 1960, and 1972, evaluates persons two years of age and older. It consists of an age-graded series of problems whose solution involves arithmetical, memory, and vocabulary skills. WS-test gives both the overall IQ as well as separate IQs for verbal and performance subtests. An example of a verbal subtest would be vocabulary breadth, while an example of a performance subtest would be picture arrangement, so that they tell a comprehensible story.

IQ was originally computed as the ratio of mental age to chronological (physical) age, multiplied by 100. If a child of 10 performs the test at the level of an average 12-year-old, this 10-year-old is considered to have a mental age of 12. In this case the child was assigned an IQ of (12/10)x100, or 120. The concept of mental age is not a persuasive one, and the computation of mental ages is not used frequently. The values of IQ are more persuasive if they are computed on the basis of statistical distributions.

Intelligence tests created a controversy about what kinds of mental abilities constitute intelligence and whether the IQ adequately represents these abilities. It turned out that intelligence tests give better results for rich kids and are worse for less privileged racial, ethnic, or social groups. Consequently, psychologists have attempted to develop culture-free tests that would more accurately reflect an individual's native ability. Johns Hopkins Perceptual Test, developed in the early 1960s for measuring the intelligence of preschool children, has a child try to match random forms (geometric forms, e.g. circles, squares, etc. are avoided because some children may be more familiar with them). Another solution was to use test materials pertinent to a child's living environment.

Psychometric tests are performed by observing and evaluating the performance of the Elementary Cognitive Tasks (ECTs) with items of ECT based on past acquired knowledge, reasoning, and problem solving requiring the concerted action of a number of relatively complex cognitive processes. A particular ECT is intended to measure a few relatively simple cognitive processes, independent of specific knowledge or information content.

Each ECT is devised to address a different set of cognitive processes, and performance on two or more different ECTs yields data from which individual differences in distinct processes can be measured, such as stimulus apprehension, discrimination, choice, visual search, scanning of short term memory (STM), and retrieval of information from long term memory (LTM). ECTs typically do not depend on previously learned information content, and in those that do, the content is so familiar that it should be common to all individuals undergoing the test.

Most ECTs are so simple that every tested individual can perform them easily. The differences in performance are measured in terms of response time (RT). The most interesting ECTs are those with RTs of less than one second and with response error rates close to zero. The subject's median RT (over n number of trials) and the subject's intraindividual variability ofRTs (measured as the standard deviation of RT over n trials) are of particular interest. Another type of ECT, known as Inspection Time (IT), measures sheer speed of perceptual discrimination (visual or auditory) independently of RT.

Measures of RT and IT derived from the various ECTs are analyzed and their correlation is estimated. For single ECTs, the correlations depend on the complexity or number of distinct processes involved in the ECT. Some processes are more strongly correlated than others. Interpretation of these correlations depends on the goal of testing and properties of intelligence that are tested.

A similar approach to testing particular skills can be exercised in the area of intelligent systems. Our ability to construct metrics should depend on the particular tools or facets of intelligence we will analyze as related to the particular performance results.

However, all psychological tests of intelligence have one feature in common: they rely upon successful performance of particular tasks, but they do not attempt to introduce any relatively comprehensive form of the model of intelligence. It is understandable for measuring intelligence of such an object as a human being. It would be unforgivable to impose similar detriment upon a researcher in cases where intelligent systems are autonomous mobile vehicles, organizational systems, large computer based control systems like unmanned power plants, structures of company management, stock market. If we succeed with these types of intelligent systems, we might be encouraged to attribute some model to a human intelligence.

PART II
RESEARCH PAPERS

PART II
RESEARCH PAPERS

1. THE PHENOMENON OF INTELLIGENCE

Using the Metaphor of Intelligence

A. Wild
Motorola, Phoenix, AZ 85018

ABSTRACT

Constructed system with autonomy can be considered as possessing intelligence, if intelligence is understood as a metaphor. It is useful to be aware of that, when defining desirable features for constructed systems, in areas such as reflecting the world (ontology), definition and pursuit of goals (teleology), or general human-like behavior (anthropomorphism). Modeling and simulating integrated systems exemplify the usage of multi-scale, multi-disciplinary representations, as a basis for increasing the autonomy of some specific constructed systems. Measuring the intelligence of constructed systems requires a Vector of Metrics for Intelligence. Its components will be defined by different means, such as conducting existence tests for essential capabilities, measuring the power to eliminate unnecessary exploration, competitions of hardware-compatible systems, or vote by a jury.

KEYWORDS: *constructed systems with autonomy, intelligence*

1. INTRODUCTION

The intelligence of the constructed systems with autonomy has to be understood as a useful metaphor, not to be stretched too far [1]. As beneficiaries of such systems, we are actually interested in their performance. The underlying assumption, however, is that building intelligence into the system, whatever its definition would be, would result in a generic and systematic way to improve their performance.

While it is relatively easy to imagine ways to measure performance, it is far less obvious how to measure intelligence, as we lack a crisp, generally accepted definition, be that for human beings, for other beings, or for artifacts.

The casual observer perceive manifestations of intelligence in multiple forms, and also will notice that somebody performing very intelligently in one situation may show what appears to be a lack of intelligence in another situation. This may suggest that intelligence is a local skill. On the other hand, some researchers intuitively feel that intelligence is an intrinsic capability of an entity, and engage in exploring the commonalties between different entities considered intelligent.

Pragmatically, the latter seems the most promising approach. If successful, it would provide the foundation for a methodology to construct systems with continuously improved capabilities. To drive the progress, it is essential to establish metrics, ranking systems according to their intelligence. Note that for this purpose it is actually irrelevant whether one considers intelligence as a generic or a local property. Depending on the viewpoint, the ranking would be valid either within a specified sub-space or in general. However, general methods, if possible, would have clearly a wider impact.

2. LIMITS OF THE METAPHOR

A multitude of aspects can be considered as elements or capabilities necessary to support intelligent behavior. In some versions, the Vector of Intelligence has 25 dimensions. It is supported by a set of computational tools, with a system architecture counting 16 features, and is completed by a control and data acquisition system with supervisory authority, also featuring a number of capabilities. Many of these elements do justice to the view adopted by the Italian Renaissance and illustrated famously by Leonardo da Vinci: the man is the measure of all things. While this approach is quite effective, and may be often unavoidable, caution is in order to avoid excesses in at least three respects: our view of the world, our goal setting capabilities and our own being.

2.1 *Ontology*

The dimensions of the vector of intelligence and the supporting tools, architectural features and auxiliary subsystem should not be excessively isomorphic with our contemporary perception of the world.

A few centuries ago, we might have asked an intelligent system to recognize the four elements and their interactions, we would have argued about the phlogiston, and hoped that eventually an intelligent system will extract the quintessence of anything and everything. It should have recognized the planets and the major stars, and have had the ability to synchronize actions with favorable skies. The Euclidean geometry was a very pertinent model to simplify the description of the world, by accepting that concepts like a straight line do have a kind of existence. Likewise, all needed knowledge about gravity was that there exists an attraction force between two bodies, precisely equal to the Cavendish constant multiplied by the two masses divided by the square of the distance. This formula easily generated the laws derived by Kepler from mountains of data and hundreds of years of observations. The depth of our understanding was made sensible (was measured ?) by this tremendous simplification.

Unfortunately, the space-time curvature of generalized relativity eliminated the paradigm of the straight line, and Newton's simple formula was unable to lead to a solution for three body interactions. Our present view is that the world does not admit a simple description.

When facing complexity, we tend to rely upon hierarchy to simplify interactions. Ideas about multi-resolution, multi-scale views imply a hierarchy. We tend to require that an intelligent system can do the same, being able to handle several hierarchy levels. Their number and their adequate utilization are candidates for intelligence metrics. Computational tools of intelligence define rules and procedures for crossing boundaries between hierarchy levels.

However common and widely accepted, the hierarchical representation of complexity is probably no more than the current model, and it seems reasonable to expect that it will be eventually replaced by a different view. This would also induce an evolution of the intelligence metrics derived from a model of the world, as it evolves historically.

As a matter of fact, the next paradigm may already take shape under our eyes: can one speak about the Internet as about a constructed system with autonomy, exhibiting intelligence ? And if yes, how would that intelligence be measured ?

2.2 *Teleology*

We consider the ability to generate goals as a leadership feature. Some philosophers consider this as the defining feature of any living beings.

However, humans, and other living creatures, pursue both explicit and implicit goals. They either conceptualized themselves the explicit goals, or receive the goals form higher authorities. In anyone of these situations, they may or may not exhibit intelligent behavior. A simple positive example is young James Watt, being given the goal to keep the pressure of a steam vessel constant. He did not conceive the goal himself, actually, he was pursuing rather different interests. It was not a goal with any recognizable intellectual challenges. But Watt generated a response that resonates until today, and will keep resonating, being, among other things, largely responsible for this workshop.

2.3 *Anthropomorphism*

A system scoring high on all dimensions of the Vector of Intelligence and its auxiliaries will probably pass easily the Turing test. It may do even more, it would be basically human, at least to the extent of our current understanding of the way humans are looking like. Some of the properties listed by Neville address the ability to communicate like humans, including such things as understanding a sentence and developing knowledge. These ideas seem to relay on the perception that the more a system is similar to a human being, the more would it be perceived as intelligent.

Even if our current understanding of humans would be definitive, this is approach may be an anthropomorphic trap. Actually, there is no necessity for the constructed structures with autonomy to present any isomorphism with our ideas about the human beings. Many of the most effective artifacts created by humankind are radically non-anthropomorphic, or non-biomorphic, for that matter. Starting with the wheel, radically different from a leg, yet allowing better locomotion, one can easily follow with any number of examples. A jet airplane is not a bird. A computer is not a brain. And a constructed automaton with autonomy is not a living being. There is no recognizable necessity for these artifacts to be indistinguishable from, or even similar to their closest living relatives.

If one recalls the number of words in any language describing non-intelligent behavior, one may conclude that copying too closely humans may be less than desirable.

3. PROGRESSING TOWARDS THE METAPHOR

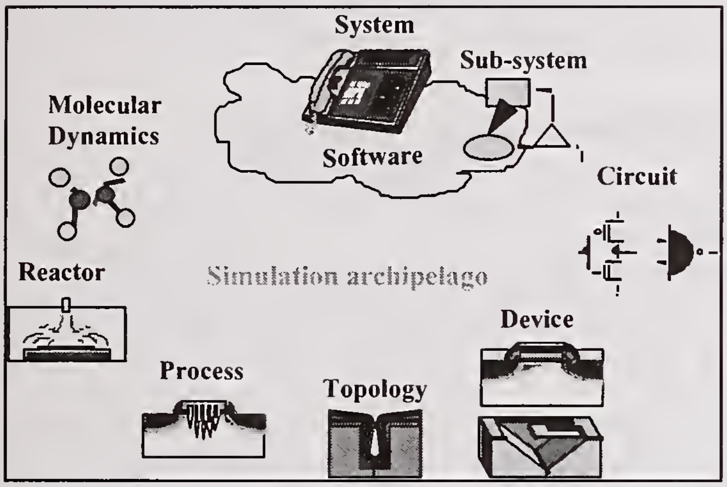
Building systems reflecting our view of the world, our purposes and our way of being, may prove productive. Multi-scale representations are probably a useful way to handle the complexity of the world in our minds, at this point in the evolution of our understanding. We can legitimately expect such representations to be useful in sciences and engineering.

The ultimate multi-discipline, multi-scale simulations are attempted by cosmologists, who hope to deduce the characteristics of the universe, 10 to 15 billion years after the Big Bang, from its characteristics when it was younger than one second.

Electronic engineers aiming to design integrated microsystems, have simpler needs: to simulate, with some quantitative accuracy, what happens on a silicon wafer within a time span from a few nanoseconds to a few hours. Microsystems are defined here as monolithic structures functionally equivalent to multi-chip systems. Increasing integration levels drive the semiconductor industry towards building system on a chip. To address this demand, design and manufacturing must integrate heterogeneous elements with traditional data processing circuits, encompassing multiple disciplines, multiple scales in space and multiple scales in time, within a coherent framework of computer aided design. Adequate modeling and simulation enables closed loop optimization and microsystem design automation.

Microsystem design must handle multi-scale modeling in time, to cope with the wide gap present in the temporal scales.

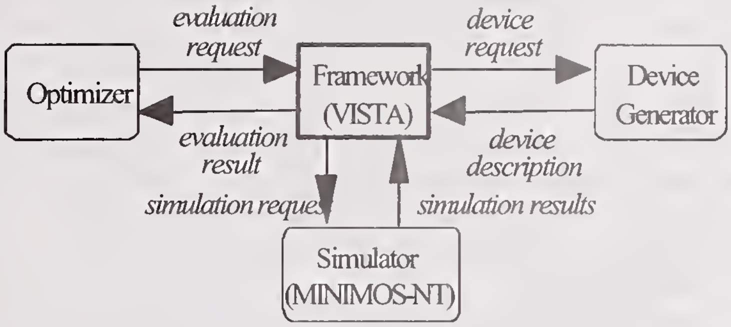
While atomistic calculations are useful for continuum simulations, molecular dynamic simulations are limited to times on the order of nanoseconds. The gap can be bridged by a meso-scale calculation, for instance using the Lattice Monte-Carlo (LMC) method to describe the hops between stable states (nanoseconds) rather than the vibration frequencies of the lattice (fractions of picoseconds). In space, multi-discipline, multi scale modeling is often required to link macroscopic reactors to microsopic integrated elements. As an example, a micromachined gear, 1 micrometer in diameter, can be analyzed using three hierarchical levels: continuum models (finite element) for the body of the wheel, molecular dynamics for gear teeth, and tight-binding for the contact between teeth. The connection is realized via a self-consistent overlap region, while keeping the time discretization in both connected domains in lock step, the whole system requiring massive parallelization at Maui Supercomputer Center.



Currently, the multiple disciplines involved in microsystems are either unconnected, building an archipelago, or put together by human programmers in an ad-hoc manner. Active research, however, is aimed at systems able to build bridges between the isolated domains, as a pre-requisite for using optimizers in closed loop. This technique allows the correlation between decisions at one manufacturing step and the system level features and performance.

Using an optimizer at the meta-level to manage the design process brings the system one step further. Many features would be required to incorporate these or similar functions in a constructed system with autonomy, exhibiting some intelligence.

This "bottom up" progression towards a development system with autonomy increasingly adds features included among the dimensions of the Vector of Intelligence. This seems a promising way towards the next challenges in engineering, believed to be nanosciences, biological systems, and last but not least, robotics. Searching for their intelligent features would surely provide underlying commonalties and accelerate the progress.



4. MEASURING THE METAPHOR

As the Vector of Intelligence and its supporting structures are multi-dimensional, multi-faceted and quite heterogeneous, a set of metrics would probably be necessary, in the hope that if a unitary definition of intelligence would emerge, a composed metric may by put forward. The four approaches presented below are the beginning of the Vector of Metrics for Intelligence.

4.1 *Counting features*

Some features of the Vector of Intelligence and the supporting structures can be tested by a go/no go test, they either exist within a given system, or they do not. Furthermore, some of them have clear numerical definitions and can be determined by counting. The result of counting is final, as long as the structure does not evolve, or represent just an assessment at that point in time, if the system can evolve. The only open problem is how to of aggregate the different dimensions of the Vector of Intelligence, so that ranking can be done.

4.2 *How far away from enumeration ?*

Testing for functional correctness of a system poses serious challenges even at the lowest levels. For example, testing the hardware of a microprocessor, a finite state machine, is conceptually easy, yet unsolvable practically. Theoretically, a test can run through all possible transitions between states, with all bit configurations at the external inputs, comparing at each step the outputs with the specification. The number of states and transitions is finite, yet so large, that the test of a 32 bit processor running at 1GHz would take a time longer that the age of the Universe.

To reduce the number of tests, one can use additional switching elements to reconfigure the structure to a finite state machine of lower complexity. If the logic gates and storage elements in the finite state machine have been defined algorithmically, one can safely accept that the functionality would be correct, if no physical defects are present. In this case, the simplified structure may be used to proof that all the desired logic gates and storage elements (a few 10 or 100 million of them on contemporary chips) are present, functional, and properly connected. These methods, currently used, are still unable to provide satisfactory test coverage. At a more abstract level, formal analysis of the structures is researched as the next opportunity to achieve it. If one adds to the testing the requirement to proof that a system or a piece of software is providing optimum responses in all cases, the complexity of the task is inhibiting.

In general, a measure of intelligence could be how much of the space to be investigated is not explored through enumeration.

This is almost isomorphic with some areas of scientific knowledge. For instance, the postulates of thermodynamics, to be accepted rather than demonstrated, point out what is impossible to achieve, saving us huge efforts, like trying to build all possible cases of perpetuum mobile of the first and second species, in addition to trying to reach absolute zero. Obviously, the postulates are very effective in eliminating an infinity of pointless attempts.

4.3 *Contests*

Intelligent systems are expected to perform well in uncertain situations, and direct competition among systems might be an appropriate way to generate uncertainty, providing means to rank them.

Examples of competitions are robot wars, fire-fighting robot contests, or robot-soccer tournaments. It is necessary to define the contests such that they address either the body or the mind of the systems in competition. Robot wars address obviously both. Athletic capabilities, rather than intelligence, also determined the outcome of the last World Cup for Robot Soccer, at which one team had access to more powerful motors than the other teams.

To dissociate the two components, an easy way would be to organize games between robots mechanically identical, but driven by different minds, a luxury seldom available with human beings.

4.4 *Vote*

Capturing all elements necessary for intelligent behavior is a complex and controversial endeavor. The Vector of Intelligence and supporting features, even after unnecessary anthropomorphic features have been eliminated, still has dimensions judged by perception.

Contemplating the behavior of living beings, one would readily identify some that would be spontaneously perceived as non-intelligent (stupid), while a whole range would be rather neutral, neither intelligent nor stupid. An alternative approach to building intelligent systems, could be to address the topic of building non-stupid systems, specifying what they should NOT do.

For instance, they should not persist in error. A non-stupid system would recognize a hopeless situation, and change its behavior or method. This distinguishes intelligence from blind instinct: ants keep building their houses even after the eggs have been removed. Although methods have been defined and implemented for quite some time to avoid stalling, quite sophisticated autonomous systems on a remote Planet still got stuck, as do soccer playing robots. When a player manages to gets unstuck by spinning, the human observers cheer. However, the opposite result is achieved, when players start spinning without a recognizable reason.

Given the subjective component in characterizing behavior as being intelligent, one could also envision scoring by the vote of a human jury. This would be similar to the methods used in some sports such as skating, in which a jury gives two notes: one for the technical merit, one for the artistic impression. After all, contests and games are entertainment, and audiences are entitled to have some fun.

5. REFERENCES

[1] White Paper of the Workshop on Performance Metrics for Intelligent Systems,
http://www.isd.mel.gov/conferences/performance_metrics

Technologies for Engineering Autonomy and Intelligence

Tariq Samad
Honeywell Technology Center
3660 Technology Drive
Minneapolis, MN 55418, U.S.A.
tariq.samad@honeywell.com

ABSTRACT

A critical need for a high performance autonomous system is the ability to generate appropriate responses when faced with conditions that were not explicitly considered during off-line design. This paper emphasizes three technical concepts as essential for meeting this need: multimodels, anytime algorithms, and dynamic resource allocation. An example from ongoing research in the autonomous uninhabited aerial vehicle domain is used to illustrate the concepts. Some competing concepts are discussed, and connections with consciousness and metrics are outlined.

Keywords: *Autonomous systems, multimodels, anytime algorithms, resource allocation, uninhabited air vehicles, consciousness.*

1. Introduction

Society, industry, and government are all exhibiting increasing interest in autonomous and semi-autonomous systems—complex engineered artifacts that require minimal or no human involvement for their operation. The motivations for this interest range from cost-efficiency to environmental safety to national defense. Potential applications are everywhere, especially where human operation is infeasible or dangerous: warfare, deep space missions, terrorism countermeasures, and toxic material handling are examples that come readily to mind.

From one perspective, it could be argued that the history of automation is the history of progress in engineering autonomy. We have been successful in automating ever-higher levels of operation, from regulatory control to supervisory control on upward. The Wright Flyer required the human pilot to perform the inner-loop control function. Today's commercial aircraft can fly from point A to point B, automatically closing the loop on not just the inner loop but also outer loop, handling qualities, and waypoint following functions.

But autonomy is much more than automation. Today's engineered systems may be highly automated, but they are brittle and capable of "hands-off" operation only under more-or-less nominal conditions. As long as the system only encounters situations that were explicitly considered during the design of its operational logic, the human element is dispensable. As soon as any abnormal situation arises, control reverts to the human.

An autonomous agent must be capable of responding appropriately to *unforeseen* situations—that is, situations unforeseen by its designers. Some degree of circumscription of a system's operating space will always exist, since survival under every environmental extreme is inconceivable, but "precompiled" behaviors and strategies are not sufficient for effective autonomy.

Below, I first discuss some features and characteristics that I believe are necessary for engineering high-performing autonomous systems. Next, in Section 3, an example from work in progress—which is focusing on the development of autonomous capabilities for uninhabited aerial vehicles—is presented. Section 4 discusses some alternative perspectives on engineering autonomy, followed by a selective review of the consciousness controversy. I conclude with a measurement-related note.

Parts of this paper are adapted from (Samad and Weyrauch, 2000) wherein some further elaboration can be found.

2. Aspects of Autonomy

What does it mean to be able to react appropriately to unforeseen situations? To be capable of exhibiting behaviors that are not precompiled? I would like to emphasize three technical concepts: multimodels, anytime algorithms, and dynamic resource allocation. These are discussed below, and a brief digression on the topic of hierarchy is also included.

2.1 Multimodels: Explicit representations of heterogeneous knowledge

In the absence of a sufficiently rich built-in library of canned responses to specific situations, an agent must be able to rely on an explicit, algorithmically manipulable knowledge base. Instead of reflexive responses being built in, the knowledge base required to generate responses deliberatively must be incorporated.

The knowledge base must capture relevant details about the capabilities of the autonomous agent, its environment, other agents it expects to be interacting with, its tasks or objectives, etc. These "models" need not be perfect; they represent what the agent believes, not objective truths. But, almost regardless of their fidelity, they allow the agent to reason and to determine responses to a potentially hostile world. The effectiveness of the responses will be a function of the fidelity of the models (in part), but, I would maintain, autonomy and effectiveness are separable. Stupid intelligence is an oxymoron; stupid autonomy is not. (In most of this paper, however, I do not make a careful distinction between intelligence and autonomy.)

I use the term multimodels to refer to multiple, heterogeneous knowledge representations. We later discuss a domain-specific example, but here I would like to note one property of multimodels that is likely to be useful across domains. The degree of precision and accuracy of knowledge that an autonomous agent must consider will vary with the situation it finds itself in. In some cases, disparate models may be used to capture different levels of detail. However, a greatly preferable option is a unified modeling framework that is capable of providing estimates or predictions at multiple levels of resolution, the level in effect at any time being specifiable by a higher level function.

2.2 Dynamic resource allocation and anytime algorithms

An autonomous agent must be able to dynamically manage its processing and other (sensing, actuation, communication, power) resources. In the face of multiple competing demands and objectives, each of which requires individual algorithmic attention, an agent cannot generally afford to examine any exhaustively. The world does not wait for closure of contemplation.

Thus, tradeoffs must be made in real-time, to decide how inevitably inadequate resources must be apportioned to the multiple demands on them. This is an issue that generally gets little attention from the intelligent systems community, yet it is no less critical than the issue of designing algorithms for information processing for autonomous systems.

Different processing tasks have different criticalities, deadlines, and other properties. Some tasks may need to be executed on a fixed periodic basis, others may be event-driven, others yet may be continually ongoing. This variety is suggestive of the complexity of real-time resource management for autonomous systems.

Of particular interest for autonomous operation are "anytime" algorithms—algorithms that are able to flexibly exploit available computational resources. Beyond a certain minimum execution time that it may require to generate an initial candidate solution, an anytime algorithm can iteratively improve on this solution over time. Randomized algorithms such as evolutionary computing are prototypical examples.

Resource management in current control systems presents an illuminating contrast with the needs for autonomous operation noted above. All control systems today have to address resource constraints. This is done by determining ahead of time—during the design process—precisely which tasks will need to be executed under what conditions. Task execution schedules can then be precomputed and defined. This static scheduling approach is infeasible for autonomous systems.

2.3 Hierarchies, but not strict ones

The sophisticated information processing systems we currently engineer are almost always hierarchical. Further, the design methodology that is proposed in today's techno-culture emphasizes strict, hierarchically structured processes. Hierarchy as an engineering design heuristic has much to recommend it, but I would assert that it is a mistake to assume that all intelligent systems must be analyzable as strictly hierarchical. One need only look at the central nervous system of any organism one thinks of as intelligent (e.g., the human brain) as evidence. There is certainly structure to the brain, but a formal, strict hierarchy is a counterfactual insistence. Bypass connections, reflex reactions, affective conditioning, many intriguing pathologies—these are all indicative of an organization that is better thought of as a web than a tree, or at least as only loosely hierarchical.

As an example, see Figure 1. Elements of the figure resemble the typical multilayer hierarchical architectures that attempts at engineering autonomous systems often adopt (i.e., the organization as shown of the spinal column, the brainstem, the thalamus, and the cerebrum). However, additional pathways are also present, forming prominent and crucial bypass structures and feedback loops.

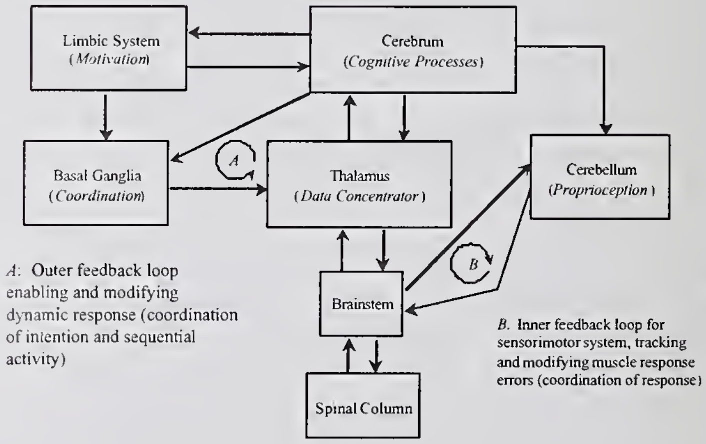

Limbic System
(Motivation)
Cerebrum
(Cognitive Processes)
Basal Ganglia
(Coordination)
A
Thalamus
(Data Concentrator)
Cerebellum
(Proprioception)
A: Outer feedback loop enabling and modifying dynamic response (coordination of intention and sequential activity)
B
Brainstem
B. Inner feedback loop for sensorimotor system, tracking and modifying muscle response errors (coordination of response)
Spinal Column

Figure 1. *Simplified architecture for primate central nervous system (figure courtesy of Blaise Morton).*

3. EXAMPLE: ROUTE OPTIMIZATION FOR AN UNINHABITED AUTONOMOUS VEHICLE

We briefly discuss here some ongoing research at Honeywell Technology Center, targeted toward the development of algorithms and software mechanisms for uninhabited air vehicles (UAVs), with specific emphasis on demanding military applications. Multimodels, anytime algorithms, and dynamic resource allocation feature prominently in our research.

An example of a multimodel knowledge base for route and trajectory optimization in a UAV is shown in Figure 2. The figure shows a (wavelet-based) multiresolution time/frequency model of a trajectory. By selectively setting specific parameters—each associated with one of the boxes in the top graphic—to zero, the space of trajectories can automatically be constrained so that different segments of the trajectory are defined in more or less detail as appropriate for a given situation. Trajectory optimization is then conducted over the enabled parameters, ensuring that computational resources are used efficiently. Under normal conditions, we can expect that the resolution profile would gradually decrease over the optimization horizon. The figure also shows multiresolution models of aircraft dynamics and terrain; these and other models are necessary to check various constraints on a hypothesized trajectory and to calculate the cost function for optimization. (See Godbole, Samad, and Gopal [2000] for more details.)

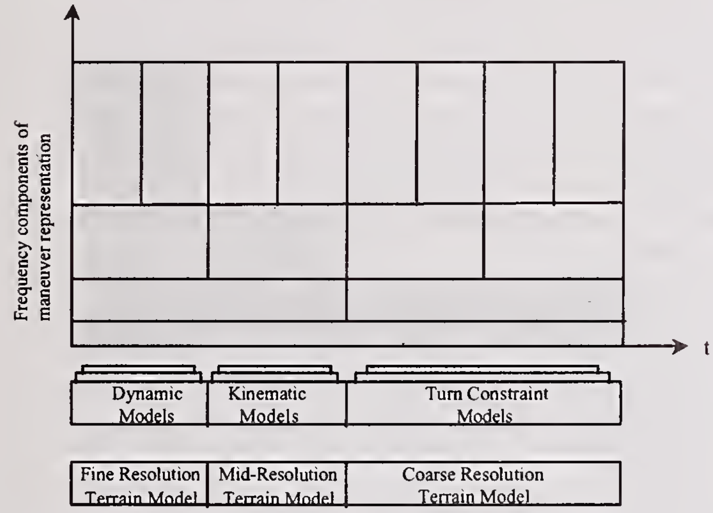

Frequency components of maneuver representation
t
Dynamic Models
Kinematic Models
Turn Constraint Models
Fine Resolution Terrain Model
Mid-Resolution Terrain Model
Coarse Resolution Terrain Model

Figure 2. *Multimodels for trajectory optimization for an autonomous aircraft.*

This multimodel approach has been integrated with an anytime algorithm for route optimization, and a simulation result is shown in Figure 3. A UAV is skirting a threat area when a target model (including the target's coordinates) is communicated to it. The original route (not shown in the figure) was not overflying the target area but instead adopting a low elevation radar-evading route over a ravine. Once the target is detected, the online trajectory optimization algorithm is executed. In this case, greater resolution is desired over a medium horizon interval, and minimizing the previous cost function for low flight is considered less important than rapidly generating an alternative route that overflies the target area. As the UAV continues its flight, incremental re-optimizations are performed at regular intervals, with the computational resources expended on these optimizations varying continuously depending on the particular objectives and models under consideration at that time.

We currently use an evolutionary computing algorithm—an extension of the algorithm outlined in (Samad and Su, 1996)—for optimizing the trajectory. The EC algorithm searches over the space of nonzero coefficients in the multiresolution wavelet-based representation noted earlier.

As I hope this example illustrates, the concepts of multimodels, anytime algorithms, and dynamic resource management are related in that effective autonomy requires the integration of all of them. Given a particular situation that requires an autonomous agent to react, it must be able:

- to access the knowledge it has that is relevant to the situation in the context of its goals and abilities;
- to flexibly reason about its decision and control options, adapting the level of scale and resolution in its processing to the situation and objectives;
- to tradeoff competing demands and requirements in the face of resource limitations.

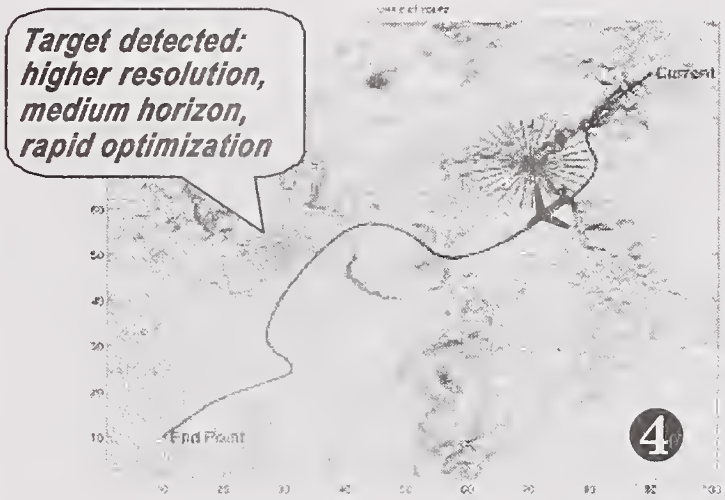


Figure 3. *A frame from a simulation example of active multimodel control for trajectory optimization.*

4. Alternative Perspectives

There are, however, other reasonable solutions and perspectives to engineering autonomy that are being proposed, and a few are briefly noted in this section.

4.1 Model-free autonomy

It seems reasonable to correlate the autonomy of a system with the fidelity or scope of the models accessible to it, a connection I have made above. The richer the explicit representations of its environment, itself, its collaborators, etc., that a system contains (regardless of whether these representations are acquired through learning or are hardwired by a designer) the more likely that an engineering system can operate effectively without continuous human supervision. So a model that can be symbolically manipulated may be seen as a necessary condition for autonomy.

But consider (as much research in intelligent systems is starting to do) an ant. There are certainly properties of ant behavior that we would be delighted to be able to incorporate within constructed systems with autonomy. An artificial ant, if we were able to construct one, would be considered to be a system with some non-trivial degree of autonomy.

Or, if the capabilities of an ant do not warrant the "autonomy" label, what about an ant colony? A million ants no more make an explicit, manipulable model of the world than an ant by itself.

The most prominent exemplar of this line of research in autonomous systems is the "subsumption architecture" of Brooks (1991), a central tenet of which is that the world can be its own model. No representations are needed—in fact, they are seen as harmful since in dynamic and ever-changing environments they can rapidly become outdated.

4.2 Is biology the only model?

Today, all the truly autonomous systems that exist are biological ones. It therefore seems appropriate to mimic salient features of biological systems in the design of engineered autonomy. However, an alternative viewpoint may lead us to question such biomimicry. Most human engineering, an endeavor that has enjoyed considerable successes, has not drawn design inspiration from biological principles—airplanes are an obvious example. Architectural sketches of brain organization (as in Figure 1) may be dismissed as irrelevant by this argument.

Of course, until some non-biologically-inspired autonomous artifact is produced, the study of existing autonomous systems (i.e., biological ones) should be helpful. But it can legitimately be argued that biology need only be a weak model.

4.3 Autonomy need not be physically grounded

Our discussion above has exemplified autonomous systems with UAVs, and most research in autonomy focuses on vehicular systems (terrestrial, undersea, or in air or space). While autonomous vehicles are a particularly exciting prospect for future engineering systems, autonomy, as a property, should not be considered constrained to physically mobile platforms.

In fact, it is important to consider autonomous systems that are not vehicles, since a broader understanding of autonomy is contingent on an understanding of the full spectrum of the topic. Different application areas will have specific characteristics. For example, in the process industries there is a continuing drive to increase the level of automation in plants, sometimes even quantified by a "number of loops per operator" metric. An autonomous decision and control system for an oil refinery will have to deal with issues related to high dimensionality (a refinery can have 20,000 sensors and actuators), significant delays due to material transport (dead times can be on the order of hours), and the lack of full state feedback.

At an even greater remove from physicality, we can contemplate autonomous computer and communication networks, which need operate only in the "virtual" realm.

5. CONSCIOUSNESS—REQUIREMENT OR RED HERRING?

The notion of developing engineered sensors or actuators, or even low-level models of computation, that are based on biologically gleaned principles is uncontroversial. Embodying higher-level cognitive capabilities in computational systems, however, is another matter. Some would argue that the sorts of phenomena found in the brains of humans cannot even in principle be realized by the sorts of machines we are contemplating. The levels of autonomy, intelligence, and adaptability exhibited by humans are thereby excluded (the argument goes) from realization in engineered systems.

The concept of consciousness lies at the center of this controversy. I take it as given that human-like performance by a machine requires the machine to have something akin to consciousness—an ability to reason about and reflect on its own behavior, not just "blindly" follow preprogrammed instructions.

There are two theoretical limitations of formal systems that are driving much of the controversy—the issue under debate is whether humans, and perhaps other animals, are not subject to these limitations. First, we know that all digital computing machines are "Turing-equivalent"—they differ in processing speeds, implementation technology, input/output media, etc., but they are all (given unlimited memory and computing time) capable of exactly the same calculations. More importantly, there are some problems that no digital computer can solve. The best known example is the halting problem—we know that it is impossible to realize a computer program that will take as input another, arbitrary, computer program and determine whether or not the program is guaranteed to always terminate.

Second, by Gödel's proof, we know that in any mathematical system of at least a minimal power there are truths that cannot be proven and falsehoods that cannot be disproved. The fact that we humans can demonstrate the incompleteness of a mathematical system has led to claims that Gödel's proof does not apply to humans.

In analyzing the ongoing debate on this topic, it is clear that a number of different critiques are being made of what we can call the "computational consciousness" research program. In order of increasing "difficulty," these include the following:

- Biological information processing is entirely analog, and analog processing is qualitatively different from digital. Thus sufficiently powerful analog computers might be able to realize autonomous systems, but digitally based computation cannot. Most researchers do not believe that analog processing overcomes the limitations of digital systems; the matter has not been proven, but the Church-Turing hypothesis (roughly, that anything computable is Turing-Machine [i.e., digitally/algorithmically] computable) is generally taken as fact. A variation of this argument, directed principally at elements of the artificial intelligence and cognitive science communities, asserts that primarily symbolic, rule-based processing cannot explain human intelligent behavior.
- Analog computers can of course be made from non-biological material, so the above argument does not rule out the possibility of engineered consciousness. Assertions that the biological substrate itself is special have also been proposed. Being constructed out of this material, neural cells can undertake some form of processing that, for example, silicon-based systems cannot. Beyond an ability to implement a level of self-reflection that, per Gödel, is ruled out for Turing machines, specifics of this "form of processing" are seldom proposed, although Penrose's hypothesis that the brain exploits quantum gravitational effects is a notable exception (Penrose, 1989). (It is worth noting that no accepted model of biological processing relies on quantum-level phenomena.)
- It has also been argued that intelligence, as exhibited by animals, is essentially tied to embodiment. Disembodied computer programs running on immobile platforms and relying on keyboards, screens, and files for their inputs and outputs, are inherently incapable of robustly managing the real world. According to this view, a necessary (not necessarily sufficient) requirement for an autonomous system is that it undertakes a formative process where it is allowed to interact with the real world.
- Finally, the ultimate argument is a variation of the vitalist one, that consciousness is something extra-material. For current purposes this can be considered a refrain of the Descartesian mind/body dualist position. Modern variations on this theme include Chalmers (1995)—an article that also includes a rebuttal by Christof Koch and Francis Crick.

The issue of consciousness in machines has captured the imagination of many as a result of the famous (or notorious) Chinese room thought experiment suggested by John Searle (1980). Searle imagines himself locked inside a room, unable to communicate with anyone outside except through slips of paper passed through a slot in the door. These slips of paper are written in Chinese, a language Searle has no knowledge or understanding of. However, Searle has been given a voluminous "script" that details (in English) the algorithmic manipulations that he should carry out upon receipt of messages. Some of the messages can have questions written on them, others may describe a story. Searle allows that the script is perfect in that the manipulations result in responses that Searle can transcribe (the symbols that he reads, manipulates, and writes are meaningless squiggles to him) and pass back to his interrogator. These responses are in fact appropriate in context; to the person outside, Searle must understand Chinese. The point of the Chinese room (thought) experiment is that knowing how the responses were generated we would not say that Searle "understands" Chinese. This is a critique of one school of thought that maintains that rule-based algorithmic processing is sufficient for understanding. Variations of the experiment and the argument have since been directed at other types of automated mechanisms.

Consciousness is a multifaceted phenomenon. I would maintain that reflective, deliberative decision making is an important element, although admittedly not the only one. Thus the technical concepts discussed earlier—multimodels, anytime algorithms, dynamic resource allocation—which, I argued, are essential for high-performance autonomous behavior, are by the same token necessary correlates of consciousness. (Our observations of) our own conscious processing support(s) this contention—we dynamically allocate cognitive resources as appropriate for an unforeseen situation, scale the precision and resolution of our processing accordingly, and rely on our knowledge of the various systems and phenomena that constitute our environment.

6. Toward Metrics

Even for humans, testing and quantifying intelligence is a controversial activity. The difficulty of compressing the multifaceted nature of intelligence into one scalar quotient has led to proposals to consider "intelligence" not as one unitary quantity but as a collection of properties that are mutually incommensurable (e.g., Gardner, 1983).

But humans, as a species, have much in common. We all have the same sensory apparatus; the same physiology, more or less; the same innate drives; the same communication apparatus; etc. If quantifying intelligence is so problematic for humans, one can wonder whether it is even sensible for artificial systems, which may have little or nothing in common. Comparing and contrasting the intelligence of an intelligent search engine for the Web with the intelligence of an autonomous vehicle is a challenge that is not only huge but perhaps unaddressable. We will need to decompose the notion of intelligence in this case too, except that instead of a handful of separate factors we might end up with a much larger number.

The technical concepts I have focused on in this paper can all be considered dimensions along which autonomy and/or intelligence can be measured. Thc extent to which an agcnt has available explicit models of relevant phenomena and systems, the scaling capabilities of the anytime algorithms available to it, and the sophistication of its adaptive computational resource allocation mechanisms, all bear on how well the agent will perform in a complex, dynamic world. More research is needed before these connections can be formalized or quantified—l have been concerned here with just their identification.

Acknowledgement

The research discussed in this paper is supported in part by the U.S. Defense Advanced Research Projects Agency (DARPA) under contract number F33615-98-C-1340.

References

Brooks, R. (1991). Intelligence without representation, *Artificial Intelligence*, vol. 47, pp. 139-159.

Chalmers, D. (1995). The puzzle of conscious experience, *Scientific American*, pp. 80-86, December.

Gardner, H. (1983). *Frames of Mind.* New York: Basic Books.

Godbole, D., T. Samad, and V. Gopal (2000). Active multi-model control for dynamic maneuver optimization of unmanned air vehicles. *Proc. Int. Conf. on Robotics and Automation*, San Francisco, CA.

Penrose, R. (1989). *The Emperor's New Mind: Concerning Computers, Minds, and the Laws of Physics*, Oxford Univ. Press.

Samad, T. and T. Su (1996). On the optimization aspects of parametrized neurocontrol design. *IEEE Transactions on Components, Packaging, and Manufacturing Technology.* vol. 19, no. 1, pp. 27-36.

Samad, T. and J. Weyrauch, eds. (2000). *Automation, Control and Complexity: An Integrated View.* Chichester, U.K.: John Wiley & Sons.

Searle, J. (1980). Minds, brains, and programs. *Bchavioral and Brain Sciences*, vol. 3, pp. 417-458.

Theoretical Constructs and Measurement of Performance and Intelligence in Intelligent Systems

Larry H. Reeker
National Institute of Standards and Technology
Gaithersburg, MD 20899
(Larry.Reeker@NIST.gov)

Abstract

This paper makes a distinction between measurement at surface and deeper levels. At the deep levels, the items measured are theoretical constructs or their attributes in scientific theories. The contention of the paper is that measurement at deeper levels gives predictions of behavior at the surface level of artifacts, rather than just comparison between the performance of artifacts, and that this predictive power is needed to develop artificial intelligence. Many theoretical constructs will overlap those in cognitive science and others will overlap ones used in different areas of computer science. Examples of other "sciences of the artificial" are given, along with several examples of where measurable constructs for intelligent systems are needed and proposals for some constructs.

Introduction

There are a number of apparent ways and certainly many more not so apparent ways to measure aspects of performance of an intelligent system. There are a variety of things to measure and metrics for doing so being proposed at this workshop, and it is important to discuss them. To develop a measure of machine intelligence that is supposed to correlate with the system's future performance capability on a larger class of tasks considered intelligent would be analogous to human IQ. That would require agreement on one or more definitions of machine intelligence and finding a set of performance tasks that can predict the abilities required by the definition(s), and still might not say much about the nature of machine intelligence or how to improve it.

One reason that metrics of performance (and perhaps, of intelligence) are needed is that they directly address the fact that it has been difficult to compare intelligent systems with one another, or to verify claims that are made for their behaviors. Another reason is that having measurements of qualities of any sort of entity provides a concrete, operational way to *define* the entity, grounding it in more than words alone. All of these aspects - *comparability*, *verifiability*, and *operational grounding* - were undoubtedly at least part of what Lord Kelvin meant about measurements providing a feeling that one understood a concept in science. (See the preamble to this workshop [Meystel *et al* 00]: "When you can measure what you are speaking about and express it in numbers, you know something about it.")

The measurements that form the primary topic of this paper are of a different type. They are ones that look ahead to the future, when the intelligent systems or artificial intelligence* field is more mature. The notion of mature field is defined here in terms of scientific theories that predict the performance of the systems on the basis of the underlying science. It is suggested that really valuable measurements require reliable predictions of this scientific sort, rather than just ways to compare the technological artifacts based on the science. To do this, it is necessary to develop theories containing measurable theoretical constructs, as will be discussed below.

The discussion of metrics for attributes of theoretical constructs herein does not conflict in any way with the idea of overall system measurements, comparisons, or benchmarks, which are useful for the purposes mentioned above. In fact, it is a philosophical problem to decide where theoretical constructs stop and empirical constructs begin. Measurements of artifacts will be referred to as **surface measurements**, those of a more theoretical nature as **deep measurements**, terms borrowed from Noam Chomsky's [65] terms for levels of syntactic description. The question of "how deep" can be left open at this time. This paper advocates looking for measurable theoretical constructs at the deeper level that will predict surface behaviors at the level of the system or subsystem, or of an entire artifact.

* The latter term will be used herein because the shortened form, "AI" is more common than "IS".

The remainder of the paper explains the form that we will expect for AI theories in the future if they are to qualify as scientific theories and suggests theoretical constructs that may have measurable properties. It will discuss existing constructs that are developing as candidates for deep metrics and how they may relate to surface measurement. It will compare them to constructs in existing scientific theories at deep and surface levels. It will suggest that they will naturally relate to constructs from the artificial and natural sciences, specifically from cognitive science and computer science.

Computation Centered and Cognition Centered Approaches to AI

At all levels, from surface to deep, the constructs to be measured may depend on the approach taken to AI. There are two distinguishable approaches that have been taken over the years, which we will call "computation centered" and "cognition centered"*. The computation centered approach focuses on how certain tasks can be accomplished by artificial systems, without any reference to how humans might do similar tasks. We do not usually think of numerical calculation as AI, but if we did, we would have to think of the way it is done as computation centered. There is no particular reason to make it cognition centered.

In the cognition-centered approach to AI, the tradition is to discover human ways of doing cognitive tasks and see how these might be done by intelligent systems. Sometimes the motivation for this approach has been to try to find plausible models for human cognitive processes (cognitive simulation), but for AI purposes, it has often been a matter of using human clues to try to accomplish the computation centered approach. Some researchers feel that developing the artifacts using cognitive ideas may lead to more robust AI systems (using "robust" in the sense that the system is not narrow or "brittle" in its intelligent capabilities). But it is a natural way to think about the developing AI capabilities, since not all areas related to intelligent activities have been explored and reduced to mathematical methods to the extent of numerical calculations, or even of mathematical logic, which might directly facilitate a computation centered approach.

Mathematical logic makes an interesting case for pointing out that most AI researchers in practice blend the computation centered and cognition centered approaches, since it is formalized, yet still can be approached in a cognition centered way. Computers actually implement mathematical logic, which is essential in control statements of programming languages. However, actually proving theorems in logic (beyond propositional logic, where truth-table methods can be used), is a creative intelligent activity. There, things become more complex, in different ways. The first complexity is that is a *creative* activity and we do not really understand even how people do it. Secondly, it is *informationally complex*: there are inherent undecidability problems in logics of sufficient richness for most interesting purposes.

In attempts to make it easier for humans to prove theorems, natural deduction methods were invented by Gentzen [34] and developed by a number of people, notably Fitch [52]. In a sense, natural deduction can be thought of as a computation-oriented version of theorem proving, taking away some of the mental work of creativity. But this does not change the inherent informational complexity problems, which provide inherent limits on computability.

Going beyond logic to general problem solving one finds some empirical studies of effective ways in which humans do it that antedate the computer. One of them, means-ends analysis, was codified in the General Problem Solver (GPS) program of Newell and Simon. [63] (See also Ernst and Newell 65]. For programs in the GPS era, it was in the spirit of that work to attempt measurement of the extent to which the program could mimic human behavior. This was done by also studying verbal protocols of people solving the problem. Any way of comparing those to the performance of the program was still pretty much a surface measurement. Such surface measures of cognitive performance, are also the heart of the Turing test [Turing 50], but do not tell us much about what is happening deeper in the system, as Joseph Weizenbaum showed with Eliza [66] (and emphasized in an ironic letter [74]). In more recent times, case-based methods have been advocated [Kolodner 88] as relating to way some people solve problems and they do look

* In the email exchange leading up to the Workshop, a third approach, "Mimetic Synthesis", whose prime concern is the "Turing test" one of representing a computer to a human user as if it were another human, was distinguished from the two mentioned by Robby Garner. It is a good distinction, though like the others, the boundaries are not always clear.

very promising. Some of the constructs from these problem-solving methods will be mentioned below.

Though computation centered and cognitive centered approaches blend well, the measurements that occur to the developers in the two approaches will naturally differ, and this is particularly true as one tries to go to a deeper level by using constructs that are based either on cognition or on computation. In other words, AI may have measurable constructs coming from at least two different sources, the computation side and the cognitive side. This fact has some interesting implications as one looks at the measurement of deeper constructs, which may have to be reconciled with both approaches to be meaningful.

The Structure of Scientific Theories

Today's views of scientific theory have changed from those held in the 19th Century, Lord Kelvin's time. The bare-bones version of a scientific theory today is that it consists of a model composed of abstract **theoretical constructs** and a **calculus** that manipulates these constructs in a way that can account for observations and accurately predict the value of experiments. The model is as central today as was the notion of measurement to Kelvin. The theoretical constructs have a relation with observed entities, properties and processes that may be quite abstract, not necessarily readily available to human senses, but following directly from calculations based on the theory. There are a number of principles applied to a model that give us increased confidence in the theory, but the one most relevant here is that we can measure the observed entities to confirm the predictions of the theories. So Kelvin's concern has been preserved, but augmented, in today's view of theories.

It is relevant to observe that the "calculus" mentioned above is used in the dictionary sense "a method of computation or calculation in a special notation (as of logic or symbolic logic)". That means that it may be numerical or non-numerical. In fact, as Herb Simon and Allen Newell [65] pointed out, there is no reason that the calculus cannot be expressed in the notation of a computer program, the better to speed its manipulation of the theoretical constructs.

For scientific theories in AI to be respectable, there will be certain requirements on them, and these affect whether they are accepted or not and whether the theories in which they occur are accepted. The late Henry Margenau had a pragmatic treatment of these requirements in his book *The Nature of Physical Reality* [Margenau 50]. A working Physicist as well as a philosopher, Margenau stressed that no amount of empirical evidence was scientifically convincing by itself, since it did not specify a unique model; and he also stressed the need for the binding of theoretical constructs to one another in a "fabric". This fabric was made up of theory and of mappings to empirical data. The theory was convincing to the degree that certain criteria were met - not a "black and white" situation, but one of degree. One of the criteria was the extent to which the models and constructs were extensible to larger and larger areas of scientific endeavor. As the fabric of the theory became larger and stronger, it became more difficult to rip it asunder.

Perhaps our emphasis on finding metrics can solidify the theoretical constructs of the field, as well as providing a means of measuring progress. The key to doing this is not to think of evaluation only as measurement of some benchmarks or physical parameters ("behaviors") that are manifested in the operation of the systems being evaluated. We need to be thinking in terms of the inner workings of the systems and how the parameters within them relate to the measured externally manifested behaviors.

One of Lord Kelvin's special interests was temperature. Temperature is of course something that we experience, something not wholly abstract. Certain physical properties are related to temperature, and the most easily observed is freezing and boiling of water. It took some scientific discovery to realize that each of these phenomena always take place at a particular (with a few reservations, like altitude and purity of the water), but still, those are concrete embodiments. Temperature has been a subjective attribute during most of the history of mankind, but the scientific notion of temperature is a theoretical construct, even though it has a close correspondence to subjective experience. The particular metrics chosen related to water boiling (in both Fahrenheit and Celsius), to Freezing (in Celsius), and to the "coldest" temperature that could be achieved with water, ice and salt (in Fahrenheit). Lord Kelvin also took the amazing step of developing a notion of temperature that is *really* abstract. His zero point of minus 273.15 degrees Celsius has never quite been reached, and is far below what any person could experience. Yet it is very real as a scientific construct, one that is part of the fabric of physical science and ties various aspects of science together in that fabric.

Many other common terms in physical theory, like mass and gravity, are theoretical constructs, though they are related to human senses. Only in relatively recent physics history have mass and gravity been understood, and we owe that understanding to bits of inspiration on the part of Galileo and Newton. Having only half a century of AI history to look back on, we cannot really expect to have such a firm fabric of theoretical constructs stitched together. But some ideas are given below, after a comparison of Sciences that study natural and the artificial systems.

Sciences of the Artificial and their relation to Natural Sciences

Herbert Simon came to the conclusion that there was a place for what he called "Sciences of the Artificial" in his important book [69]. He did not *invent* the study of artifacts in a systematic manner, but he realized accurately and acutely that that artifacts could be subjects of "real sciences", with deep theories of the sort that exist in natural sciences. We will now consider some of the implications of this idea.

The boundaries between sciences of the artificial and the natural sciences are not clear-cut in practice because nature colors human artifacts, determining their possibility and their features. The "engineering sciences", the portions of engineering that has been formalized in the sense of that they can predict the behavior of artifacts, including aspects such as stability and strength can be considered sciences of the artificial. The reason that this is not remarked upon more often is that they have called upon physical sciences more and more over the centuries to aid the "ingenuity" that gives the profession its name.

Linguistics is a science of the artificial. Human language is the artifact that it studies. But of course, the properties of the artifact are shaped by the natural properties of human learning and cognition, human hearing and speech in many ways. In the domain of phonetics, for example David Stampe's "natural phonology" [Stampe 73, Donegan and Stampe 79] characterizes some of the interactions between language as an artifact and as a natural phenomenon. We do not understand even yet the extent of the interaction between linguistics and human cognition. Is there an LAD (language acquisition device) [Chomsky 75] innate in humans that is specific to language, or is the learning of language based on the same principles as such other acquired systems as visual perception? Nobody knows for sure; but whatever the case, the nature of the world and the nature of learning processes must affect language.

Computer Science is a science of the artificial. Certainly, this is true insofar as it studies computers, which are artifacts; but also to the extent that it studies algorithms, which are human creations, too. The main subject studied in much of Computer Science is not computers but information, and the "state", which is all the relevant information about a system at a given time, is therefore a fundamental theoretical construct. Information is a theoretical construct that is also fundamental in the natural sciences, but whose significance as a theoretical construct has only become apparent in this century, as its relationship to entropy and its role in quantum theory have been realized. So again, Computer Science has both artificial and natural parts.

Economics, another science of the artificial, studies a major artifact, the economy, and looking at this science of the artificial can provide some insight into the position of AI as a science of the artificial, and of the role of measurable theoretical constructs.

Predictive Measurement in a Science of the Artificial – An Example from Economics

Economics has struggled for longer than AI has existed to find theoretical constructs that have predictive power. Economics deals with large amounts of aggregated data, so its empirical data are statistical in nature, and its models are not as clear as physical models with respect to the interrelationships among theoretical constructs, nor are they as widely accepted. Yet they do allow some prediction of economic performance and are used in control processes for the purposes of economic stability, with a degree of success.

As this paper is being written, the U.S. Federal Reserve Bank is aggressively raising interest rates because the *employment rate* (inferred from job creation and unemployment data) is high and *economic growth* (a function of GDP change and other data) has been rapid. In their models, this predicts increasing *inflation* (as measured by the *consumer and producer price indices* and other constructs). It has recently been conjectured that there should be a role in these models for *productivity*, the role of which is not yet fully understood. So economic theory, as it develops, must relate all of these constructs and others: average interest rates, supply and demand for money and goods, savings rate, etc.

Economic theories and their constructs are still complex and incomplete. Incorporated in complex computer models, their predictions are not totally trustworthy, but the predictions are testable. Economics provides an example from another science of the artificial that AI should follow in formulating and measuring constructs.

Surface Measures and Theoretical Constructs in AI – Some Examples

The sort of predictive ability that economists want, we would like to see in AI, too. If we have theoretical constructs at some deeper level, we can also use the theories of which they are a part to simulate or predict mathematically what happens if we increase or decrease parameters related to those constructs. It is a thesis of this paper that *there are theoretical constructs that can predict system performance measured in terms of surface measures.* At this point in the development of AI as science, it is hard to say just exactly what they would be, but some ideas can be drawn from today's AI and related subjects.

An Example Construct: Robustness

A surface measurement that could be very valuable across a variety of systems is some measure of *robustness* – the ability to exercise intelligent behavior over a large number of tasks and situations. From a computation-centered standpoint, if systems become robust, AI progress would be easier to see. From a cognition-centered standpoint, a system can never really be intelligent if it is not robust. (One way to think of a measure of intelligence in a single system would be as a measure of performance, robustness and autonomy.) The *surface* way to determine the robustness of a system would be to try it on a number of tasks and see how broad its methods are. But what *makes* intelligent systems robust? Learning ability, experience, and the ability to transfer that experience to new situations are all things that come to mind. A rough sketch of how measuring theoretical constructs in those areas might give us a predictive figure for developing robust systems is given below.

Robustness: Learning?

If learning can make systems more robust, it should be interesting to measure the strength of the system's learning component. How easily does it adapt the system to a new situation? *Unsupervised learning* has wide applicability, but it can basically only determine clusters of similar items. *Supervised learning* must be presented with exemplars to learn relations, which seems not to be enough for a machine to extend its own capabilities. *Reinforcement learning* (RL) is a blend of both cognitive and computational centered AI. It started out as a model of classical conditioning, but turned out to be applied dynamic programming. There are a number of different techniques within RL, all of which have many possible applications. Neural nets or other approaches may be used. The theoretical constructs include the *state space* chosen, the *reinforcement function*, and the *policy*. The field is becoming quite sophisticated, and there are known facts about the relation of these to outcomes in particular cases [Mahadevan and Kaelbling 96]. Suppose that a reinforcement learning system constitutes a part of the intelligence of an intelligent system. There should be some way of predicting how that system would do upon encountering problems of a certain nature. By knowing how it chooses the concepts in its system and how they react on problems of that type, one can provide a partial evaluation of how effective the learning system would be. By obtaining such figures for all such subsystems, one could relate them to the performance of the full intelligent system. There is much work to be done in that direction.

Under certain circumstances, one can imagine learning extending robustness; but having to learn each new variations of a problem, even by reinforcement, is unlikely to lead to robustness quickly. It is expected that reinforced behaviors learned in one situation might be identical to those needed in another system, so this may lead to more rapid or better learning in the second situation. One approach to this is to condition behaviors that are not built into the system initially, as explored by Touretzky and Saksida [97]. But, still, one would like to have more general ways of reusing "big pieces" of learned knowledge.

Robustness: Transfer of Learning?

Transfer of learning is a phenomenon that we may be able abstract to theoretical constructs that can help to predict robustness. It is still not a deep measure, so it will then be important to predict transfer of learning from deeper constructs which will be mentioned below. At present, it is a research challenge to build transfer of learning into systems. But it is possible to see how one could test for it.

As far as measurement, here is roughly how transfer of learning might be measured:

1. Machine performance is measured on Task 1. The score is P(t1, T1)) = performance at time t1 on Task 1. P is some suitably broad performance measure.
2. Performance is measured on Task 2 without learning (this being an artifact where we can control learning) to obtain P(t1, T2) (keeping the time variable the same because the same machine abilities are assumed without learning even if the measurements are not simultaneous).
3. Note that if the measure is to have a meaning, previous training that might affect T1 or T2 must be controlled for, which could be difficult.
4. The machine is now allowed to perform task T1 in which it learns, achieving better performance at some time t2, i.e. P (t2, T1) > P (t1, T1).
5. It is then tested on T2, and the question is whether P (t2, T2) > P (t1, T2) without having done additional learning on Task 2.

If indeed P (t2, T2) > P(t1, T2) in some quantifiable way, the system has achieved (at least locally) one of the goals of AI, the transfer of learning from T1 to T2. The amount of transfer can be measured by the amount of improvement on task2 as a function of the amount of training on task T1. Let us assume that we can describe this by some transfer effectiveness function, E for the system being tested. Let us say E(T1, T2, t) gives "the effectiveness of training on T1 for time t in terms of transfer toT2". We could describe this by a graph of performance on T2 as a function of time being spent on T1.

Developing such a measure of transfer of learning and getting it accepted is not simple. To be useful, we would need a way of comparing T1 and T2, to be sure that the second task is not just a subtask to the first. Difficult or not, defined measurements such as these are steps toward understands the construct "transfer of learning" and achieving it in artifacts. The measurable transfer construct would, in turn, help to provide a measurement of robustness, since learning transfer can make a system more robust. It is a step toward measurement of intelligence, at least by some definitions of intelligence, and, intuitively, at least, would have some predictive power.

How might we go about defining the similarity of T1 and T2, as suggested above? We would have to decide what we mean by similarity of task. An interesting essay in this area is "Ontology of Tasks and Methods" [Chandrasekaran, Josephson and Benjamins [98]].

Various candidates for potentially measurable constructs that could be used to produce transfer but also to relate transfer to other phenomena are mentioned in a book edited by Thrun and Pratt [98], who have both had a research interest in learning-transfer processes. From the computation side comes the possibility of changing *inductive bias*. From the cognition-centered side, there is *generalization* from things already learned; but *overgeneralization* can be a major problem in learning, so it needs to be constrained. (Some simple constraints on overgeneralization in language learning are discussed in [Reeker 76].)

Robustness: Case-Based Reasoning?

Case-based reasoning is an intuitively appealing technique that was mentioned earlier in this paper. The idea is that one learns an expanding set of cases and stores the essentials of them away according to their conventional features. They are then retrieved when a similar case arises and mapped into the current case. Potential theoretical constructs include *indexing* and *retrieval* methods for the cases, case *evaluation* and case *adaptation* to the new situation. The cases could also be abstracted and generalized to various degrees, to a *model.*

Case-based reasoning is important for cognition centered AI. It is intuitively the way many people often figure out how to do things, and is thus embodied in the teaching methods of many professional fields – law, business, medicine, etc. It provides a launching pad for creativity as well, as mappings take place from one case to an entirely new one. Perhaps the new case is not really concrete, but a vague new idea. Then the mapping of an old case to it may result in a creative act – what we usually call *analogy*. Analogy, *metaphor* in language, is a rich source – absolutely ubiquitous – of new meanings for words, and thus of new ways to describe concepts, objects, actions. Perhaps one key to robustness is the ability to use analogy. Four interesting papers by researcher in the area can be found in an issue of *American Psychologist* [Gentner *et al* 97].

Existing Surface and Subsurface Performance Measures

Researchers in text-based information retrieval (IR) have traditionally considered themselves not to be a part of the AI field, and some have even considered that artificial intelligence was a rival technology to theirs; but there is an overlap of interest. It is worth noting that IR has had a useful surface measure of system performance that has guided research and allowed comparison of technologies. The measure consists of two numbers, *recall* and *precision* [Salton 71]. Recall measures the completeness of the retrieval process (the percentage of the relevant documents retrieved). Precision measures the purity of the retrieval (the percentage of retrieved documents judged relevant by the people making the queries). If both numbers were 100%, all relevant documents in a collection would be retrieved and none of the irrelevant ones. Generally, techniques that increase one of the measures decrease the other. Real progress in the general case is achieved if one can be increased without decreasing the other.

For the IR community, better recall and precision numbers have both shown the progress of the field. They also show that it is still falling short, keeping up the challenge, especially as the need to use it for very large information corpora rises. In addition, they provide a standard within the community for judging various alternative schemes. Given a particular text corpus, one can consider various weighting schemes, use of a thesaurus, use of grammatical parsing that seeks to label the corpus as to parts of speech, etc., to improve the retrieval process. The interesting thing is to relate these methods and the characteristics of the corpus to precision and recall, but so far that has not been sharp enough to quantify generally.

Related to information retrieval is automated natural language information extraction, which tries to find specified types of information in bodies of text (often to create formatted databases where extracted information can be retrieved or mined more readily). A related but different (cost-based) measure was defined several years ago for a successful information extraction project [Reeker, Zamora and Blower 83]. One measure was *robustness* (over the texts, not different tasks as in the broader intelligent systems usage discussed earlier). This was defined as the percentage of documents out of a large collection that could be handled automatically. The idea was that some documents would be eliminated through automated pre-screening (because those documents were not described by the discourse model the system used) and relegated to human processing. Another measure was *accuracy* (the percentage of documents not eliminated that were then correctly processed in their entirety, by the system). Yet another was *error rate* (the percentage of information items that were erroneous – including omitted - in incorrectly handled documents). From this more detailed breakdown, estimates of the basic cost of processing the documents, based on human and machine processing costs and costs assigned to errors and omissions, was derived. The measure could be used to drive improvements in information extraction systems or decide whether to use them, compared to human extraction (which also has errors) or to improve the discourse model to handle a larger portion.

For information extraction projects, it was further suggested that the cost of erroneous inputs might drive a built-in "safety factor" that could be varied for a given application [Reeker 85]. This safety factor was based on linguistic measures of the text (in addition to the discourse model) that could cause problems for the system being studied. The adjustable safety factor could be built into the prescreening mentioned above. In other words, the system would process autonomously to a greater or lesser degree and could invite human interaction in applications where the cost of errors was especially high. It was suggested that the system would place "warning flags" to help it make a decision on screening out the document, and these could also aid the human involved. Although this was a tentative piece of work, the idea of tying a surface measure (robustness) into the underlying properties of the system is exactly like tying measurable surface properties into underlying theoretical constructs. The theoretical constructs mentioned in this case were structural or semantic ones from linguistics.

From the area of software engineering comes another tradeoff measure that is worth mention. The author did some work on ways of providing metrics - surface metrics, initially - for program readability (or understandability) [Reeker, 79]. Briefly, studies of program understanding had identified both go-to statements and large numbers of identifiers (including program labels) as problems. At the same time, the more localized loop statements could result in deep embeddings that were also difficult to understand for software repair or modification. The vague concept of readability could be replaced by a measure of go-to statements and maybe also one of the number of different identifiers. This particular study suggested *depth of embedding* as a problem and also suggested a tradeoff between depth of embedding a metric called *identifier load.* Identifier load was a function of the number of identifiers and the span of program statements over which they were used. Identifier load tended to increase as depth of embedding was reduced by the obvious methods.

There were a number of similar software metrics studies in the 1970s, and they continue. This approach, however, was part of an attempt to look at natural language for constructs that might be of relevance in programming languages and programming practice [Reeker 80]. The *depth* measure was based on an idea of Victor Yngve [60], which came out of his work in linguistics - an idea that retains a germ of intuitive truth. Yngve had in turn related his natural language measure of embedding depth to measures of short-term memory from cognitive psychology. Whether these relationships turn out to be true or lead to related ideas that are true or not, they illustrate how theoretical constructs can stitch AI, computer science, and other artificial and natural sciences together. They also illustrate the quest for metrics that can firm up the foundations of the sciences.

More Constructs To Be Explored

There are many more existing theoretical constructs that have arisen within AI or been imported from computer science or cognitive science that beg to be better defined, quantified, and related to other constructs, both deep and surface.

Means-ends analysis and *case based reasoning* have both been mentioned as forms of problem solving. How do these cognitive characterizations of problem solving relate to one another? At a deeper level is the construct of *short term memory* mentioned in the previous section in relationship to Yngve's *depth.* How does short-term or working memory relate to long term memory and how are the two used in problem solving? The details are not known. The size of a short-term memory may not be as relevant in a machine, where memory is cheap and fast. But we cannot be sure that it is not relevant to various aspects of machine performance because it is reflected at least in the human artifacts that the machine may encounter. For instance, in resolving anaphora in natural language the problem may be complicated if possible referents are retrieved from arbitrarily long distances.

A similar problem arises from long-term memory if everything ever learned about a concept is retrieved each time the concept is searched for. This can lower retrieval precision (to use the term discussed earlier for machine retrieval) and cause processing difficulties on a given problem. It may be that Simon's notion of *bounded rationality* is a virtue in employing intelligence. Are we losing an important parameter in intelligence if we try always to optimize rationality? For AI system, *anytime algorithms* and similar constructs for approximate, uncertain, and resource bounded reasoning have been developed in recent years, and hold a good deal of promise [Zilberstein 96].

An interesting theoretical construct arising out of AI knowledge representation and the attempts to use it in expert systems and agents and for other purposes is that of an *ontology.* "Ontology" is an old word in philosophy designating an area of study. In AI it has come to designate a type of artifact in an intelligent system: The way that that system characterizes knowledge. In humans, ontologies are shared to a large degree, but certainly differ from every person to every other, despite the fact that we can understand each other. Are some ontologies indicative of more intelligence than others in ways that we can measure? One suggested criterion for high intelligence is the ability to understand and use very fine distinctions (or to actually create new ones, as described in Godel's memorandum cited by Chandrasekaran and Reeker [74]). Is an ontology's size important, or its organization, or both? Can one quantify a system's ability to add new distinctions?

A related issue is *vocabulary.* Many people think that an extensive vocabulary, *used appropriately*, is a sign of intelligence, or at least

Characterizing Three Related Endeavors Involving Computers and Intelligence and Their Purposes[2]

Name of Endeavor	Mimetic Synthesis	Cognitive Modeling	Artificial Intelligence
Principal Goal	Produce behavior that appears to be evidence of cognition or intelligent phenomena	Produce models of cognitive processes, including learning, planning, resoning, perception, linguistic behavior, etc.	Find ways of doing with computers things that we deem intelligent when they are done by humans.
Use	Produce illusion of intelligent behavior for interface purposes, entertainment, etc.	Develop psychological theories of cognition. [1]	Tools to augment intelligence and systems that exhibit increasingly intelligent behavior autonomously.
Category of Endeavor	Computer Technology	Psychological Science (Branch of Natural Science)	Computer Science (Branch of Science of Artificial)
Approach	Use simulations, stored answers, AI or cognitive models, or other techniques that are convincing to human users.	Use evidence from psychological experiments; make working models; test against human behavior.	Use techniques from mathematical and engineering disciplines, cognitive models, and previous experience. Test through programs.
Examples	Eliza (J. Weizenbaum), Albert (Garner and Henderson), Talking Coke Machines (?),…	LT, GPS (Simon and Newell), HAM (J. Anderson), SOAR (A. Newell),…	Deep Blue (IBM), SATPLAN (Kautz and Selman), Dendral (Buchanan, Feigenbaum..), TD-Gammon (Tesauro),…

[1] By this we mean the traditional cognitive psychology level, not brain function. The latter is a biological approach. In the nature of science, of course, one expects theories of such close areas to be consistent and to inform one another, and to merge in the longer term.

[2] Clearly, each of these endeavors is different, though each can make use of knowledge from the others and some devices could solve all goals. The confusion between them, however, has resulted in misunderstandings for decades.

scholastic aptitude. In computer programs that do human language processing, the vocabulary consists of a *lexicon* that generally also has structural (*syntactic*) information for parsing or generating utterances containing the lexical item and *meaning representations* for the lexical item. The lexicon can be much larger than any human's vocabulary; but for the vocabulary to be used appropriately for language production or understanding, it still falls far short of the human vocabulary. For that to be improved better techniques of *semantic mapping* are required, including links to ontologies and methods of inferring the ontological connections and of idiosyncratic aspects of speakers with which a conversation is taking place. Is the vocabulary an indication of the size of the ontology and the distinctions it makes, or vice-versa? Nobody knows; but better theories of how they link up are needed for both understanding and fully effective use of human language by intelligent systems.

Another cognitive cóncept that is still a mystery is *creativity*, certainly a part of intelligence, or at least of high intelligence. Does the ability to add entirely new concepts, not taught, constitute creativity? How does one harness serendipity to develop creativity? Is creativity linked with *sensory cognition*, the cognitive phenomena related to senses, such as vision, including perception, visual reasoning, etc. There is a need for deep theoretical constructs underlying notions like creativity, and for measures of these constructs and their attributes [Simon 95, Buchanan 00].

Turning to computational constructs, we notice that much of the AI described above takes place through various forms of *search*. Already there exists a pretty good catalogue of variations on search and how to manage it, in which a good deal of theory is latent. Some of the search is of a *state space*, involving the ubiquitous state concept basic to theoretical computer science. Search is also coupled with *pattern matching*, which underlies many of the methods mentioned earlier in this paper.

The potential constructs mentioned here are just a sample of the ones already available in Artificial Intelligence, and to them should be added others found in some of the major works of Newell and Simon on Problem Solving and Cognition [Newell and Simon [65], Newell [87]].

Summary and Author's Note

The development of a true science of artificial intelligence is something that has concerned the author for a long time. It has been encouraging to see the development within the field of interesting and non-obvious theoretical constructs. This paper has suggested that theoretical constructs with attributes that we can measure are especially valuable and it has suggested a number of such candidates. The paper suggests that we enlist Lord Kelvin's emphasis on measurement in choosing such constructs. These same measurable theoretical constructs will in many cases relate (at least at deeper levels) to those of cognitive science, computer science, and other sciences. They will help predict measures at the surface that can be used to provide metrics for the performance (and through that, the intelligence) of intelligent artifacts. We should have in mind the quest for such measurable constructs as we move forward in creating intelligent artifacts.

References

Buchanan, B. G. [00], Creativity at the Meta-Level, Presidential Address, American Association for Artificial Intelligence, August 2000. [Forthcoming in *AI Magazine.*]

Chandresekaran, B., J. R. Josephson and V. R. Benjamins [98] Ontology of Tasks and Methods, 1998 Banff Knowledge Acquisition Workshop. [Revised Version appears as two papers "What are ontologies and why do we need them?," *IEEE Intelligent Systems*, Jan/Feb 1999, 14(1); pp. 20-26; "Ontology of Task and Methods," *IEEE Intelligent Systems*, May/June, 1999.]

Chandrasekaran, B. and L. H. Reeker [74]. "Artificial Intelligence – A Case for Agnosticism," *IEEE Trans. Systems, Man and Cybernetics*, January 1974, Vol. SMC-4, pp. 88-94.

Chomsky, Noam [65]. *Aspects of the Theory of Syntax*. MIT Press, Cambridge MA.

Chomsky, N. [75] *Reflections on Language*. Random House, New York.

Donegan, P. J. & D. Stampe [79]. The study of Natural Phonology. In Dinnsen, Daniel A. (ed.*)*. *Current Approaches to Phonological Theory*. Indiana University Press, Bloomington, 126-173.

Ernst, G. & Newell, A. [69]. *GPS: A Case Study in Generality and Problem Solving*. Academic Press, New York.

Fitch, F. B. [52]. *Symbolic Logic.* Roland Press, New York.

Gentner, D., K. Holyoak *et al* [97]. Reasoning and learning by analogy. (A section containing this introduction and other papers by these authors and A. B. Markman, P. Thagard, and J. Kolodner, *American Psychologist*, 52(1), 32-66.)

Gentzen, G. [34] Investigations into logical deduction. *The Collected Papers of Gerhard Gentzen*, M. E. Szabo, ed. North--Holland, Amsterdam, 1969. [Published in German in 1934.]

Kolodner, J.L. [88] Extending Problem Solving Capabilities Through Case-Based Inference, In, *Proceedings of the DARPA Case-Based Reasoning Workshop*, Kolodner, J.L. (Ed.), Morgan Kaufmann, Menlo Park, CA.

Leake, D. B. Ed. [96] *Case-Based Reasoning: Experiences, Lessons, And Future Directions* Indiana University, Editor 1996, AAAI Press/MIT Press, Cambridge, MA.

Margenau, H. [50]. *The Nature of Physical Reality*, McGraw-Hill, New York.

Mahadevan, S. and L. P. Kaelbling [96] The National Science Foundation Workshop on Reinforcement Learning. *AI Magazine* 17(4): 89-93.

Meystel, A. *et al* [00] Measuring Performance of Systems with Autonomy: Metrics for Intelligence of Constructed Systems. *In this volume.*

Newell, A. [87] *Unified Theories of Cognition.* Harvard University Press. Cambridge, Massachusetts, 1990. [Materials from William James Lectures delivered at Harvard in 1987.]

Newell, A., and H.A. Simon [63] GPS, a program that simulates human thought, *Computers and Thought*, E.A. Feigenbaum and J. Feldman (Eds.), McGraw-Hill, New York.

Newell, A., & Simon, H.A. [65]. Programs as theories of higher mental processes. R.W. Stacy and B. Waxman (Eds.), *Computers in biomedical research* (Vol. II, Chap. 6). Academic Press, New York.

Newell A. & Simon H.A. [72] *Human Problem Solving*, Prentice-Hall, Englewood Cliffs, NJ.

Reeker, L. The computational study of language acquisition, *Advances in Computers*, 15 (M. Yovits, ed.), Academic Press, 181-237, 1976.

Reeker, L. [79] Natural Language Devices for Programming Language Readability; Embedding and Identifier Load, *Proceedings, Australian Computer Science Conference*, Hobart Tasmania.

Reeker, L., E. Zamora and P. Blower [83] Specialized Information Extraction: Automatic Chemical Reaction Coding from English Descriptions, *Proceedings of the Symposium on Applied Natural Language Processing,* Santa Monics, CA, Association for Computational Linguistics, 1983.

Reeker, L. H. [80] Natural Language Programming and Natural Programming Languages, *Australian Computer Journal* 12(3): 89-93.

Reeker, L. H. [85] Specialized information extraction from natural language texts: The "Safety Factor", *Proceedings of the 1985 Conference on Intelligent Systems and Machines*, 318-323, Oakland University, Michigan, 1985.

Salton, G. [71] *The SMART Retrieval System*, Prentice-Hall, Englewood Cliffs, NJ.

Simon, H. A. [69] *The Sciences of the Artificial.* Third Edition. Cambridge, MA, MIT Press, 1996. [Original version published 1969].

Simon, H. A. [95] Explaining the Ineffable: AI on the Topics of Intuition, Insight and Inspiration. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, IJCAI 95, Montréal, Morgan Kaufmann, Menlo Park, CA, Volume 1, 939-949.

Stampe, D. [73] *A Dissertation on Natural Phonology.* New York: Garland Publishing, 1979. [Original University of Chicago dissertation submitted in 1973.]

Thrun, S. and L. Pratt (eds.) [98]. *Learning To Learn.* Kluwer Academic Publishers.

D.S. Touretzky and L.M. Saksida [97] Operant conditioning in Skinnerbots. *Adaptive Behavior* 5(3/4):219-247.

Turing, A. M. [50]. "Computing Machinery and Intelligence." *Mind* LIX (236 ; Oct. 1950): 433-460 reprint in [*Collected Works of A. M. Turing* vol. 3: Mechanical Intelligence, D. C. Ince ed., Elsevier Science Publishers, Amsterdam, 1992: 133-160].

Weizenbaum J [66]. ELIZA - a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9, 36-45.

Weizenbaum, J. [74]. Automating psychotherapy. *Communications of the ACM*, 17(7):425. July 1974.

Yngve, V [60] The depth hypothesis, *Proceedings, Symposia in Applied Mathematics*, Vol. 12: Providence, RI, American Math. Society, 1961. [Based on publication under another title, 1960.]

Zilberstein, S. [96] Using Anytime Algorithms in Intelligent Systems, *AI Magazine*, 17(3):73-83, 1996.

Intelligence with Attitude

W. C. Stirling and R. L. Frost
Electrical and Computer Engineering Department
Brigham Young University
Provo, UT 84602

ABSTRACT

An essential feature of intelligence is the ability to make autonomous choices. A new paradigm of satisficing decision making incorporates two utilities for decision making, rather than the usual single utility that is characteristic of optimal decision making. These two utilities may be used to define figures of merit for the intellectual power or fitness of the decision maker as it functions in its environment. These utilities may also be applied in group settings. In particular, societies of negotiatory decision makers may undergo considerable tension as they attempt to reach a compromise that is acceptable to the group as a whole and to all members of the group.

KEYWORDS: *multi-agent decision theory, satisficing, attitude, negotiation*

1. INTRODUCTION

There are three issues that must be addressed in the design of an intelligent decision system: (a) defining the alternatives, (b) defining the preferences, and (c) choosing between the alternatives as a function of the preferences. The first two issues are highly dynamic. Alternatives may appear and disappear and preferences may change. Much of the study of intelligent systems is properly focused on these dynamics. At the moment of truth when a decision must be made, however, we must assume that the alternatives and preferences have been defined, and all that remains is to make the choice. This paper focuses on this last, consummate step.

The ability to make decisions is essential to intelligent behavior. Indeed, the word *intelligent* comes from the Latin roots *inter* (between) + *legere* (to choose). We thus assume that there is only one essential characteristic of intelligence in man or machine—an ability to choose between alternatives.

Choices between alternatives, or decisions, are usually justified by the maximization of expected utility, an approach Simon calls *substantive rationality* [8]. We argue that for multiple agents, especially those in dynamic environments, the requirement for substantive rationality is too demanding. First, although a solution may exist, the information or computing power necessary to find it may be unavailable. We will often be forced to fall back on what Simon terms *procedural rationality*, or the reliance on heuristic or *ad hoc* procedures defined by an authority. Second, and more serious, is that the existence of an optimal solution may be in doubt. Von Neumann-Morgenstern game theory shows that for many games a solution that is simultaneously best for the group and for each individual in the group simply does not exist. This seems to imply that a theory of group decisions satisfactory for the synthesis of coordinating agents cannot be obtained by a straightforward maximization of utility.

We are thus motivated to consider definitions of rationality upon which we can build a more robust theory of intelligent multi-agent decision making. We hold that the fundamental obligation of a rational decision maker is to make decisions that are, in some well-defined sense, good enough. Historically, the study of good enough decisions was first formalized by Simon, when he introduced the term *satisficing* to characterize decisions that achieve the decision maker's *aspiration level* [6, 7]. This notion of satisficing defines quality according to the criteria used for substantive rationality, but evaluates quality against a standard that is chosen more or less arbitrarily. It essentially blends substantive and procedural rationality, and is a species of what is often termed *bounded rationality*.

Rather than blend the two extremes of substantive and procedural rationality *a la* Simon, our work explores an alternative which leads naturally to a set of satisficing solutions that is consistent with Simon's intent. It also guarantees the existence of jointly rational decisions, and seems to be a natural vehicle for the design and synthesis of intelligent decision systems.

We start by assuming that the most primitive way to make decisions is to make intra-option comparisons in the form of dichotomies. We define two distinct (and perhaps conflicting) sets of attributes for each option and to either select or reject the option on the basis of comparing these attributes. Such dichotomous comparisons are *intrinsic*, since the evaluation of an option's merits is not referenced to anything not directly related to the option, including other options. They are also local comparisons; it is not possible to form a global ordering the options on the basis of such comparisons. An *intrinsically rational* choice is one for which the decision maker's benefits are at least as great as its costs. We define a *satisficing decision* as one that is intrinsically rational,[1] because these options are good enough, in the sense that their attributes have been favorably compared with a standard. We differ from Simon only in the standard used for comparison: the positive and negative attributes of each option, versus externally supplied aspiration levels.

Intrinsic rationality appears to be a weaker notion than substantive rationality. Although it identifies all options that are, in the sense we have defined, good enough, it does not insist on a unique solution. At the moment of truth, the decision maker may choose any of the satisficing options with the assurance that it will at least get its "money's worth." In practice, however, the advantage of a theory founded on substantive rationality may be more illusory than real. Objective functions themselves are often created by an *ad hoc* combination of preferences into a single performance index, and this combination can be, and usually is, manipulated until satisfactory behavior is achieved. Thus, even optimization approaches rely in their application on satisficing notions, however informally.

As mentioned earlier, our approach to intrinsic rationality requires the definition of two preference functions, one to characterize the desirable attributes, and one the undesirable attributes, of each option. An option is desirable to the degree that it achieves the goal. It is undesirable to the degree to which its adoption consumes the decision maker's resources, such as energy, safety, or other costs. Separate preference functions permit the development of metrics to evaluate how suited the decision maker is to function in its environment. Intuitively, if a decision maker has options available to it that achieve its goal with low cost, it is well-suited for its environment. On the other hand, if it must incur great cost or undergo great risk to achieve its goal, it is clearly not as well suited. Although the goal may be achieved equally well in either case, there is a fundamental difference in the ability of the agent under the two scenarios. This difference may not be easily discernible under the substantive or procedural rationality paradigms, but it is clearly discernible under the intrinsic rationality paradigm.

In the following we first summarize the mathematical development of satisficing decision theory. We next introduce a concept of *attitude*, or disposition, for the agents, and develop figures of merit for evaluating the equivocation experienced by the decision maker or decision making system. We then present a basic negotiation theorem and describe a simple negotiatory process to converge to a rational compromise. We then finish with an example and draw conclusions.

2. SATISFICING

Von Neumann-Morgenstern game theory is based on a very sophisticated paradigm—global optimization. There are a number of basic problems, however, with optimization-based approaches. First, since it is well known that humans are not good optimizers [1, 2, 5], a decision-making system that seeks to approximate human behavior may be unnecessarily constrained by insisting on, and only on, optimal performance. Second, optimization is a fixed, or absolute concept, in the sense that if an option is not the best, then it is unacceptable. There cannot be degrees of optimization. Third, optimization is, fundamentally, a notion of exclusive self interest, and does not easily generalize to settings where it is important to accommodate both group and individual interests [4]. It is usually impossible to arrive at a joint solution that is simultaneously best for the group as a whole and for each member of the group.

Our notion of satisficing, on the other hand, does not insist upon optimal performance, and in return for this concession it logically permits degrees of satisficing and the accommodation of both group and individual interests. By adjusting the tradeoff standards between cost and benefit, it may be possible to find a joint solution that is simultaneously good enough for the group and good enough for each member of the group. This is the fundamental goal of negotiation.

Our approach is to employ the mathematics, but not the usual semantics, of probability theory. As discussed in [9, 10] we may encode the preference relationships via mass functions, which we term the *selectability* and *rejectability* functions. By so doing, we are able to account for conditional preferences (analogous to conditional probabilities) and to express both joint (group) and marginal (individual) preferences.

We formalize this procedure as follows. Let U_i denote the option set for the ith agent (we will assume U_i is of finite cardinality), $i = 1, \ldots, N$, let $\mathbf{U} = U_1 \times \cdots \times U_N$ denote the product space of joint options, and let $\mathbf{u} = \{u_1, \ldots, u_N\}$, where $u_i \in U_i$, denote an option vector. Let $p_{\mathbf{S}}(\mathbf{u})$ indicate the degree to which the joint option $\mathbf{u}$ is successful in achieving a group goal. We require that $\sum_{\mathbf{u}\in\mathbf{U}} p_{\mathbf{S}}(\mathbf{u}) = 1$ and $p_{\mathbf{S}}(\mathbf{u}) \geq 0$, so $p_{\mathbf{S}}$ is a mass function, which we term the *joint selectability mass function*. Also, let $p_{\mathbf{R}}(\mathbf{u})$ indicate the degree to which the joint option $\mathbf{u}$ consumes resources, and require this to also be a mass function, which we will term the *joint rejectability mass function*. Next, let $p_{S_i}\colon U_i \to [0,1]$ and $p_{R_i}\colon U_i \to [0,1]$ be marginal selectability and rejectability mass functions, respectively, derived from $p_{\mathbf{S}}$ and $p_{\mathbf{R}}$ by appropriate summation. For a discussion of how these joint and marginal mass functions may be practically constructed, see [9, 10].

These mass functions define a dichotomy for each option, that is, they partition the attributes of the option into two categories and provide a measure of support for each class of attributes. We evaluate each dichotomy by comparing the selectability (benefit) to the rejectability (cost) of each option. By so doing, we define the *jointly satisficing set*

$$\mathbf{\Sigma}_b = \{\mathbf{u} \in \mathbf{U}\colon p_{\mathbf{S}}(\mathbf{u}) \geq b p_{\mathbf{R}}(\mathbf{u})\},$$

and define the *individually satisficing sets*

$$\Sigma_b^i = \{u \in U_i\colon p_{S_i}(u) \geq b p_{R_i}(u)\},$$

[1] Other researchers have appropriated this term to describe various notions of constrained optimization. In this paper, we restrict our usage to be consistent with Simon's original concept.

$i = 1, \ldots, N$. The *boldness parameter, b*, is a constant in the interval $[0, 1]$, which is nominally set to unity, but may be decreased under special circumstances to be discussed below. $\mathbf{\Sigma}_b$ is the set of all joint options that are good enough for the group, and each Σ_b^i is the set of all individual options that are good enough for the ith agent.

These sets provide the agent or group of agents with the ability to make individual or group decisions. If the ith individual agent is empowered to make its own decision, it may choose any member of Σ_b^i. If the group as a whole is to make a collective decision, it may choose any member of $\mathbf{\Sigma}_b$. These choices may be random, or they may be made according to some tie-breaking procedure.

3. EQUIVOCATION

Human decision makers often make qualitative assessments of the difficulty, in terms of stress or tension, encountered in making decisions. Even if such knowledge does not have a direct bearing on their immediate decisions, an appreciation of the difficulty involved in forming the decision is an important aspect of the decision-making experience. A decision maker need not possess anthropomorphic qualities, however, to assess the difficulty of making decisions, and we do not propose to endow an artificial decision maker with some sort of ersatz anthropomorphic capability. Under our satisficing approach, however, it is possible to evaluate attributes of the decision problem that correspond more to its functionality and fitness than to its success.

Are decisions easily made and implemented, or do they tax the capabilities of the decision maker? Such assessments are not a typical undertaking of classical decision theory. Maximizing expectations has no need to concern itself with issues such as "difficulty." Nevertheless, choices are not all of equal difficulty.

By employing two utilities, rather than only one, we may analyze them to ascertain the compatibility of the attributes of the preferences. If they are compatible, in that options that conserve resources also achieve the goal, then the decision maker is in a fortunate situation of being content. If the preferences are incompatible, in that options that achieve the goal also are highly consuming of resources, then the decision maker is fundamentally conflicted. These attributes constitute attitudes, or dispositions, of the decision maker.

The optimization literature is devoid of discussions concerning the attitude or disposition of the decision maker who, like the paradigm it employs, is assumed to be dispassionate. It is simply doing what should be done under the auspices of individual rationality, and attitudes or feelings, should they even exist (and they need not), are completely irrelevant. Furthermore, to attribute anthropomorphic characteristics to a decision maker would be seen by many as nothing more than a concocted story line that is of marginal value if not completely misleading.

3.1. *Attitude*

It is fortunate if an option that conserves resources (low rejectability) also achieves the goal (high selectability)—in this environment, a decision maker is content. Many interesting decision problems, however, are such that actions taken in the interest of achieving the goal are expensive, hazardous, or have other undesirable side effects. A decision maker in this situation is conflicted. Contentment and conflict are basic dispositional states that serve as guides to the decision maker's functionality. A situation requiring frequent high-conflict decisions indicates that the tasks are difficult for the decision maker. Making high-conflict decisions, however, is not a measure of how well the decision maker is performing—it may, in fact, be making good, but costly, decisions. It is also true, however, that a high-conflict environment may result in poor performance because the decision maker is simply not powerful enough to deal adequately with its environment. Such a situation might serve as a trigger to prompt changes, such as activating additional sensors, or otherwise seeking more information about the environment. It may also trigger a learning mechanism to prompt the decision maker to adapt itself better to the environment.

Since selectability and rejectability are probabilities, it may be useful to appropriate some of the mathematical machinery of probability theory to aid in interpreting these quantities. One way to gain some insight is to examine the entropy of selectability and rejectability.

Definition 1 The **entropy** of a mass function p is

$$H(p) = -\sum_{u \in U} p(u) \log_2 p(u).$$

□

Entropy is usually employed in Shannon information theory as a measure of how much uncertainty (randomness or disorder) is reduced, on average, as a result of conducting an experiment governed by the mass function [3]. In our context, however, we wish to provide entropic interpretations for selectability and rejectability that are distinct from the usual probabilistic interpretation.

In assessing selectability, we consider expediency as analogous to uncertainty. To motivate this interpretation, suppose u' is implemented. If $p_S(u') \approx 1$, then $\log_2 p_S(u') \approx 0$ which is consistent with the notion that little reduction in expediency occurs if an option with high selectability is implemented. Conversely, suppose $p_S(u') \approx 0$, but is nevertheless implemented. Then $-\log_2 p_S(u')$ is large, indicating a great loss in expediency. The entropy of selectability is the average reduction in expediency that obtains as result of making choices according to p_S.

To interpret the entropy of p_R, we consider expense as analogous to uncertainty. Suppose u' is implemented. If $p_R(u') \approx 1$, then $\log_2 p_R(u') \approx 0$ which is consistent with the notion that little reduction in expense occurs if a highly rejectable option is nevertheless implemented. On the other hand, if $p_R(u') \approx 0$ and u' is implemented, then $-\log_2 p_R(u')$ is large, indicating a great reduction in expense. The entropy of rejectability is the average reduction in expense that obtains as a result of making choices according to p_R.

Entropy is maximized by the uniform distribution; that is, if $p^*(u) = \frac{1}{n}$ for all $u \in U$, then $H(p^*) \geq H(p)$ for all mass functions p over U, and has entropy $H(p^*) = \log_2 n$. A uniform p_S generates the highest possible average expediency, and a uniform p_R would generate the highest possible average expense. Consequently, it is useful to take the uniform distribution as a baseline against which to assess the properties of arbitrary mass functions. Let n be the cardinality of the action space, U (assumed to be finite for this discussion).

Definition 2 If $p_S(u) = \frac{1}{n}$ (that is, selectability under p_S is equal to selectability under the uniform distribution), then the option is **success neutral**. If the selectability mass function is uniform, then the decision maker's attitude will be success neutral. □

Definition 3 If $p_R(u) = \frac{1}{n}$ (that is, rejectability under p_R is equal to rejectability under the uniform distribution), then the option is **conservation neutral**. If the rejectability mass function is uniform, then the decision maker's attitude will be conservation neutral. □

Definition 4 If $p_S(u) > \frac{1}{n}$ (that is, selectability under p_S is greater than selectability under the uniform distribution), then the option is attractive with respect to performance relative to other options—u is **expedient**. □

Definition 5 If $p_R(u) > \frac{1}{n}$ (that is, rejectability under p_R is greater than rejectability under the uniform distribution), then u is unattractive with respect to cost or other penalty–u is **expensive**. □

The relationship between selectability and rejectability permits the definition of four dispositional modes of the decision maker with respect to each of its options. Let U be the set of all possible options.

Definition 6 If $u \in U$ is both expedient and expensive, then the decision maker will desire to reject, on the basis of cost, an option that is suitable in terms of performance—it will be **ambivalent** with respect to u. □

Definition 7 If $u \in U$ is both inexpedient ($p_S(u) < \frac{1}{n}$) and inexpensive ($p_R(u) < \frac{1}{n}$), then the decision maker will be desirous of accepting the option on the basis of cost, but will be reluctant to do so because of poor performance. The decision maker will be **dubious** with respect to u. □

Definition 8 If $u \in U$ is expedient and inexpensive, then the decision maker is in the position of desiring to implement an option that would yield good performance—a dispositional mode of **gratification** with respect to u. □

Definition 9 If $u \in U$ is inexpedient and expensive, then the decision maker will desire to reject, on the basis of cost, an option that also provides poor performance, and will thus be in a dispositional mode of **relief** with respect to u. □

These four modes provide a qualitative measure of the way the decision maker is matched to its task. Gratification and relief are modes of contentment, while dubiety and ambivalence are modes of conflict. Figure 1 illustrates these regions.

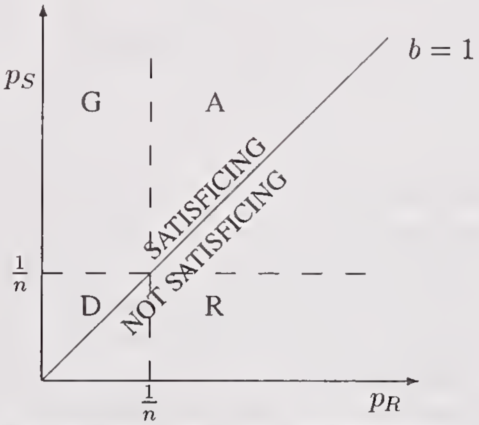
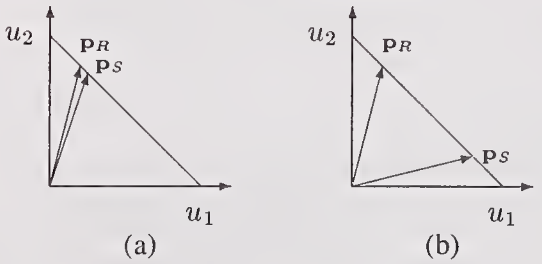


Figure 1: Dispositional regions: G = gratification, A = ambivalence, D = dubiety, R = relief.

Figure 2 illustrates various cases for $n = 2$, a two-dimensional decision problem. In these plots, the diagonal line represents the unit simplex, and the p_S and p_R values are plotted as vectors that lie on the simplex.



(a)

(b)

Figure 2: Attitude: (a) The decision maker is dubious with respect to u_1 and ambivalent with respect to u_2. (b) The decision maker is gratified with respect to u_1 and relieved with respect to u_2.

3.2. *Figures of Merit*

It would be useful to obtain formal expressions to capture some of the features of the qualitative analysis described in Section

3.1., where it is qualitatively indicated that as these distributions become more closely aligned, the decision maker becomes more ambivalent and dubious. We propose two measures that are similar, but not identical.

Diversity One important feature of the selectability and rejectability functions, therefore, is their dissimilarity. To obtain such a measure, we again appeal to the notion of entropy, and apply the Kulback-Leibler distance measure.

Definition 10 The **Kulback-Leibler (KL) distance measure** of two mass functions, say p_1 and p_2, is given by

$$D(p_1 \parallel p_2) = \sum_{u \in U} p_1(u) \log_2 \frac{p_1(u)}{p_2(u)}.$$

□

The KL distance measure is an indication of the relative entropy of two mass functions. $D(\cdot \parallel \cdot)$ is not a true metric; it is not symmetric and does not obey the triangle inequality. It is, however, non-negative, and it is easily seen that $D(p_1 \parallel p_2) = 0$ if and only if $p_1(u) = p_2(u)$ for all $u \in U$.

We may apply the KL distance measure to the problem of ascertaining dissimilarity of the selectability and rejectability functions by computing the KL distance between selectability and rejectability.

Definition 11 The **diversity functional** is:

$$D(p_S \| p_R) = \sum_{u \in U} p_S(u) \log_2 \frac{p_S(u)}{p_R(u)},$$

or, equivalently,

$$D(p_S \| p_R) = -\sum_{u \in U} p_S(u) \log_2 p_R(u) - H(p_S).$$

□

Small values occur when the selectability and rejectability functions are similar, indicating a condition of potential conflict. If they are identical, then the decision maker is in a position of wishing to reject precisely the options that are in its best interest—an unfortunate condition of total paralysis.

Diversity is infinite if there exist options with nonzero selectability and zero rejectability. Such options are free options, since no cost independent of achieving the goal is incurred by adopting them (analogy: coasting saves fuel, but may or may not get you to your destination). Diversity is not a measure of performance; that is, if one decision maker has a more diverse selectability/rejectability pair than another, that is not an indication that it will perform better than the other. It does, however, provide an assessment of the environment in which the decision maker operates.

Tension Although the diversity functional provides insight into the relationship between selectability and rejectability, it does not afford a convenient comparison in the case where the decision maker is neutral with respect to either selectability or rejectability. To develop such a measure, it is convenient to re-normalize the selectability and rejectability functions. Consider first the case where p_S and p_R are mass functions and U is finite. Let

$$\begin{aligned} \mathbf{p}_S &= [p_S(u_1), \ldots, p_S(u_n)] \\ \mathbf{p}_R &= [p_R(u_1), \ldots, p_R(u_n)] \end{aligned}$$

be selectability and rejectability vectors, and let $\mu = [\frac{1}{n}, \ldots, \frac{1}{n}]$ denote the uniform mass function vector, where n is the cardinality of U. Although these vectors are unit-length under the L_1 norm, they are not of unit length under the L_2 norm. It will be convenient to normalize these vectors with respect to L_2. Let $|\mathbf{p}_S| = \sqrt{\mathbf{p}_S \mathbf{p}_S^T}$, with similar definitions for $|\mathbf{p}_R|$ and $|\mu|$. The L_2 normalized mass function vectors will be denoted by $\tilde{\mathbf{p}}_S = \frac{\mathbf{p}_S}{|\mathbf{p}_S|}$, and similarly for $\tilde{\mathbf{p}}_R$ and μ.

We express the similarity between p_S and p_R through the inner product of the corresponding unit vectors, yielding the expression $\tilde{\mathbf{p}}_S \tilde{\mathbf{p}}_R^T$. This quantity will be unity when $p_S \equiv p_R$, and will decrease as the two mass functions tend toward becoming orthogonal, and thus captures some of the properties we desire to model. If we normalize by the product of the projections of $\mathbf{p}_S$ and $\mathbf{p}_R$ onto the uniform distribution, we tend to scale up the inner product as the mass function vectors become distanced from the uniform distribution.

Definition 12 The **tension functional** is

$$T(p_S \| p_R) = \frac{\tilde{\mathbf{p}}_S \tilde{\mathbf{p}}_R^T}{\tilde{\mathbf{p}}_S \tilde{\mu}^T \tilde{\mathbf{p}}_R \tilde{\mu}^T},$$

which simplifies into the convenient form:

$$T(p_S \| p_R) = n \mathbf{p}_S \mathbf{p}_R^T = n \sum_{i=1}^{n} p_S(u_i) p_R(u_i).$$

□

Clearly, $T(p_S \| p_R)$ is positive and bounded by the dimension, n. If either the selectability or rejectability is uniform, then the tension function equals unity. If the rejectability function is uniform, then the decision maker is rejectability-neutral. If the selectability is uniform, then the decision maker is selectability-neutral. If $T(p_S \| p_R) > 1$, then the projection of selectability onto rejectability is significant, and options that are desirable are also costly. We may interpret this as a state of conflict. On the other hand, f $T(p_S \| p_R) < 1$, then the projection of selectability onto rejectability is small, and the decision maker is in a state of contentment.

A decision maker operating in a contented environment is well-tuned to its task—decisions that possess high rejectability also possess low selectability. Such a decision maker should be expected to achieve its goals with ease, and be adequate in most situations. A conservation-neutral decision maker will function much as would a conventional Bayesian decision-maker. If it is success-neutral, it will function much like a minimax decision-maker. If the decision maker is both conservation-neutral and success-neutral, it is completely indifferent to the outcome, and there is little point in even attempting to make a decision other than a purely random guess.

4. NEGOTIATION

Negotiation under the individual rationality paradigm forbids any individual participant, as well as any potential coalition, from settling for a decision that is below its security, or minimax, level. This is a very strong restriction, which can lead to an empty core and the lack of a rational basis for negotiation. There are many ways to modify this solution concept to justify solutions not in the core, such as accounting for bargaining power based on what a participant calculates it contributes to a coalition by joining it (e.g., the Shapley value), or forming coalitions on the basis of no player having a justified objection against any other member of the coalition (e.g., the bargaining set). Also, it is certainly possible to invoke various voting or auctioning protocols to address this problem. We do not criticize the rationale behind these refinements to the basic theory, or the various extra-game-theoretical considerations that may govern the formation of coalitions, such as friendship, habits, fairness, etc. We simply point out that to achieve a reasonable solution it may be necessary to go beyond the strict notion of maximizing individual expectations and employ ancillary assumptions that temper the attitude and behavior of the decision makers

Satisficing negotiation, however, permits controlled degrees of altruism. If agents are willing to lower their standards, as defined by the boldness, b, they may obtain a satisficing compromise, where a joint decision is obtained that is good enough for the group as a whole and good enough for each member of the group. This potential result is guaranteed by the following theorem.

Theorem 1 *(The* **negotiation theorem**.*) If u_i is individually satisficing for the ith agent, that is, $u_i \in \Sigma_b^i$, then it must be the ith element of some jointly satisficing vector $\mathbf{u} \in \mathbf{\Sigma}_b$.*

Proof We will establish the contrapositive, namely, that if u_i is not the ith element of any $\mathbf{u} \in \mathbf{\Sigma}_b$, then $u_i \notin \Sigma_b^i$. Without loss of generality, let $i = 1$. By hypothesis, $p_{\mathbf{S}}(u_1, \mathbf{v}) < b p_{\mathbf{R}}(u_1, \mathbf{v})$ for all $\mathbf{v} \in U_2 \times \cdots \times U_N$, so $p_{S_1}(u_1) = \sum_{\mathbf{v}} p_{\mathbf{S}}(u_1, \mathbf{v}) < b \sum_{\mathbf{v}} p_{\mathbf{R}}(u_1, \mathbf{v}) = b p_{R_1}(u_1)$, hence $u_1 \notin \Sigma_b^1$. □

The content of the negotiation theorem is that, under intrinsic satisficing, no one is ever completely frozen out of a deal—every decision maker has, from its own perspective, a seat at the negotiating table. This is perhaps the weakest condition under which negotiations are possible.

A decision maker possessing a modest degree of altruism would be willing to undergo some degree of self-sacrifice in the interest of others. Such a decision maker may be viewed as an **enlightened liberal**; that is, one who is intent upon pursuing its own self interest but gives some deference to the interests of the group in general. Such a decision maker would be willing to lower its standards, at least somewhat and in a controlled way, if doing so would be of great benefit to others or to the group in general.

The natural way for a decision maker to express a lowering of its standards is to decrease its boldness. Nominally, we may set b_i, the boldness of the ith agent, to unity, which reflects equal weighting of the desire for success and the desire to conserve resources. By decreasing b_i, the agent lowers its standard for success relative to resource consumption, and thereby increases the size of its satisficing set. As $b_i \to 0$ the standard is lowered to nothing, and eventually every option is satisficing. Consequently, if all decision makers are willing to reduce their standards sufficiently, a compromise can be achieved.

Figure 3 illustrates this negotiatory process. The amount by which b_i must be reduced below unity is a measure of the degree of compromising needed to reach a mutually acceptable solution. As with tension and diversity, however, this degree of compromising is not a measure of performance, but it is a useful figure of merit for assessing the degree of difficulty that is associated with the negotiatory process.

Step 1: Agent i forms $\mathbf{\Sigma}_{b_L}^i$ and $\Sigma_{b_i}^i$, $i = 1, \ldots, N$; initialize with $b_i = 1$, $b_L = \min\{b_1, \ldots, b_N\}$.

Step 2: Agent i forms its compromise set by eliminating all option vectors for which its component is not individually satisficing, resulting in $\mathbf{C}_i = \{\mathbf{u} \in \mathbf{\Sigma}_{b_L}^i : u_i \in \Sigma_{b_i}^i\}$.

Step 3: Broadcast $\mathbf{C}_i$ and b_i to all other participants, receiving similar information from them.

Step 4: Form the satisficing imputation set, $\mathbf{N} = \cap_{j=1}^N \mathbf{C}_j$. If $\mathbf{N} = \emptyset$, then decrement b_j, $j = 1, \ldots, N$, and repeat previous steps until $\mathbf{N} \neq \emptyset$.

Step 5: Agent i implements the ith component of the rational compromise

$$\mathbf{u}^* = \arg\max_{\mathbf{u} \in \mathbf{N}} \frac{p_{S_1 \cdots S_N}(\mathbf{u})}{p_{R_1 \cdots R_N}(\mathbf{u})}.$$

Figure 3: The Enlightened Liberals negotiation algorithm.

This leads to a theory of social behavior than is very different from standard N-person von Neumann-Morgenstern game theory. Whereas, under conventional theory, additional criteria may be required to foster successful negotiations, the satisficing concept builds controlled degrees of compromise into the decision-making procedure. If an agent reaches its limit of compromise before negotiations are successful, it may be forced to declare an impasse, rather than to sacrifice its standards any further.

5. RESOURCE SHARING

The following simple example illustrates the fundamental differences between substantive and intrinsic rationality. Suppose a factory operates N processing stations that function independently of each other, except that, if their power requirements exceed a fixed threshold, they must draw auxiliary power from a common source. Unfortunately, there are only $N - 1$ taps to this auxiliary source, so one of the stations must operate without that extra benefit. Although each station is interested in its individual welfare, it is also interested in the overall welfare of the factory and is not opposed to making a reasonable compromise in the interest of overall corporate success.

Let U denote the set of auxiliary power levels that are feasible for each X_i to tap, and let $f_i \colon U \to [0, \infty)$ be an objective function for X_i; that is, the larger f_i, the more effectively X_i achieves its goal. X_i's choice is tempered, however, by the total cost of power, as governed by an anti-objective function, $g_i \colon U \to [0, \infty)$, such that the smaller g_i, the less the cost. Work cannot begin until all players agree on a way to apportion the auxiliary power. Table 1 displays these quantities for a situation involving three decision makers.

U	f_1	g_1	f_2	g_2	f_3	g_3
0.0	0.50	1.0	0.10	1.0	0.25	1.0
1.0	2.00	2.0	2.00	3.0	0.50	5.0
2.0	3.00	4.0	3.00	6.0	1.00	5.0
3.0	4.00	5.0	4.00	9.0	2.00	5.0

Table 1: The objective functions for the Resource Sharing game.

A standard approach under substantive rationality is to view this as a cooperative game. The payoffs may be obtained by combining the two objective functions, yielding individual payoff functions of, say, the form

$$\pi_i(u_1, u_2, u_3) = \begin{cases} -1 & \text{if } u_j > 0 \, \forall j \\ \alpha_i f_i(u_i) - \beta_i g_i(u_i) & \text{otherwise} \end{cases},$$

$i = 1, 2, 3$, where α_i, β_i, and μ are chosen to ensure compatible units. To achieve this compatibility, we normalize f_i and g_i to unity by setting $\alpha_i = \frac{1}{\sum_{u \in U} f_i(u)}$ and $\beta_i = \frac{1}{\sum_{u \in U} g_i(u)}$.

The Pareto solution is $\mathbf{u}_P = \{0, 1, 3\}$, but, with an attitude governed by expected utility maximization, X_1 has no incentive to agree to this apportionment. Thus, to solve this problem, a negotiation protocol must be invoked. Of the various protocols that are possible, the only one that does not require assumptions additional to that of self-interested expectations maximization is the core. Unfortunately, the core is empty for this game. Essentially, this is because only two decision makers can share in the auxiliary power source, effectively disenfranchising the third decision maker. This situation potentially leads to an unending round of recontracting, where participants continually make offers and counter offers in a fruitless attempt for all to maximize their expectations.

Let us now view the decision makers in their true character as enlightened liberals who are willing to accept solutions that are serviceably good enough for both the group and the individuals. From the point of view of the group, an option is satisficing the joint selectability exceeds the joint rejectability scaled by boldness. We define joint rejectability as the normalized product of the individual costs functions, namely,

$$p_{R_1R_2R_3}(u_1, u_2, u_3) \propto g_1(u_1) g_2(u_2) g_3(u_3),$$

where "$\propto$" means the function has been normalized to sum to unity. To compute the joint selectability, we note that, under the constraints of the problem, only two of the agents may use the auxiliary power source. We may express this constraint by defining the joint selectability function as

$$p_{S_1S_2S_3}(u_1, u_2, u_3) \propto \begin{cases} p_{S_1}(u_1) p_{S_2}(u_2) p_{S_3}(u_3) & \text{if } \mathbf{u} \in \Pi \\ 0 & \text{otherwise} \end{cases}$$

where Π is the set of all triples $\mathbf{u} = \{u_1, u_2, u_3\}$ such that exactly one of the entries is zero. The individual rejectability and selectability marginal mass functions are obtained by summing over these joint mass functions according to the rules of probability theory.

The enlightened liberals algorithm yields, for $b > 0.8$, an empty satisficing imputation set. But, when b is decremented to 0.8, the satisficing imputation set is $\mathbf{N} = \{\{0, 1, 3\}, \{0, 2, 3\}, \{0, 3, 3\}\}$ and the rational compromise is $\mathbf{u}^* = \{0, 1, 3\}$ which, coincidentally, is the Pareto optimal solution. It is not surprising that, at unity boldness, there are no options that are simultaneously jointly and individually satisficing for all participants, since there is a conflict of interest (recall that the core is empty). But, if each individual adopts the point of view offered by intrinsic rationality, it gradually lowers its personal standards to a point where it is willing to be content with reduced benefit, provided its costs are reduced commensurately, in the interest of the group achieving a collective goal. The amount b must be reduced to reach a jointly satisficing solution is an indication of the difficulty experienced by the participants as they attempt to resolve their conflicts. Reducing boldness is a gradual mechanism for decision makers to subordinate individual interest to group interest. This mechanism is very natural in the regime of making acceptable tradeoffs, but is quite foreign to the concept of maximizing expectations ("you get what you pay for" versus "nothing but the best").

The diversity and tension values for this decision problem are given in Table 2. We interpret these values as follows.

Agent	Diversity	Tension
X_1	0.55	0.93
X_2	0.03	1.30
X_3	1.21	0.73
Group	2.85	0.51

Table 2: Diversity and Tension for Resource Sharing Game.

Group diversity is high and group tension is low, indicating that, as a group, the system is fairly well suited for its environment, and that the system is powerful enough to make good decisions. Individually, X_2 has the lowest diversity and the highest tension. This situation is reflected in the structure of N, where we see that X_2 has several choices that are good enough, but is either dubious or ambivalent about all of them. Thus, X_2 experiences the most conflict in making decisions. X_3 is quite content with its decision and so is X_1. The fact that X_1 is not conflicted as measured by diversity and tenseness may appear somewhat contradictory, since it is X_1 who ends up sacrificing for the benefit of the group. But these figures of merit are not intended to be metrics of performance, only of the intellectual power of the decision maker, in terms of its conflict between selectability and rejectability.

6. CONCLUSION

An intelligent agent is, first and foremost, a decision maker, regardless of the problem context, the way knowledge is represented, or the criteria used to define performance. One way to assess the functionality of the agent is to provide it with a means to evaluate introspectively its own fitness, or suitability, to function in its environment. Satisficing decision theory provides this capability. Although the figures of merit associated with these fitness evaluations are not measures of performance, they are useful measures of the innate intellectual (decision-making) power of the agent.

References

[1] M. Bazerman. A critical look at the rationality of negotiator judgement. *Behavorial Science*, 27:211–228, 1983.

[2] M. H. Bazerman and M. A. Neale. Negotiator rationality and negotiator cognition: The interactive roles of prescriptive and descriptive research. In P. H. Young, editor, *Negotation Analysis*, pages 109–129. Univ. of Michigan Press, Ann Arbor, MI, 1992.

[3] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley, New York, 1991.

[4] R. D. Luce and H. Raiffa. *Games and Decisions*. John Wiley, New York, 1957.

[5] A. Rapoport and C. Orwant. Experimental games: a review. *Behavorial Science*, 7:1–36, 1962.

[6] H. A. Simon. A behavioral model of rational choice. *Quart. J. Econ.*, 59:99–118, 1955.

[7] H. A. Simon. Rational choice and the structure of the environment. *Psychological Review*, 63(2):129–138, 1956.

[8] H. A. Simon. Rationality in psychology and economics. In R. M. Hogarth and M. W. Reder, editors, *Rational Choice*. Univ. Chicago Press, Chicago, 1986.

[9] W. C. Stirling and M. A. Goodrich. Satisficing games. *Information Sciences*, 114:255–280, March 1999.

[10] W. C. Stirling, M. A. Goodrich, and D. J. Packard. Satisficing equilibria: A non-classical approach to games and decisions. *Autonomous Agents and Multi-Agent Systems Journal*, 2000. To appear.

What is the Value of Intelligence?

Thomas Whalen
Professor of Decision Sciences, Department of Management
Robinson College of Business, Georgia State University
whalen@gsu.edu

Abstract

Probably the most widespread and significant existing "performance metric for intelligent systems" is the dollar premiums that employers are willing to pay to recruit and retain more intelligent human employees compared to less intelligent ones. This paper examines some of the aspects driving this economic metric in the search for analogies that may be useful in establishing performance metrics for constructed intelligent systems. Aspects considered include Language Understanding & Capacity to Act, Goal-Directedness, Autonomy and Unpredictability, Information, Uncertainty, World Models, and Self-Models and Self Awareness. The paper concludes with a discussion of performance metrics for human intelligence and a brief prospectus for the role of economic considerations in assessing the Vector of Intelligence

Keywords: *economic value, intelligence*

1. Introduction

Much of the discussion leading up to the conference on "Performance Metrics for Intelligent Systems" focuses on an "inner" view of intelligent performance, or rather of intelligence itself. This inner view takes two very different forms: components like memory or MIPS that must be present inside an intelligent system, and metaphysical questions about the "inner life" of an intelligent system, such as questions of consciousness.

Rather than try directly to add to this interesting and valuable train of thought, this paper approaches the subject of performance metrics for intelligent systems from an external perspective. The question under consideration hers is "What is the economic value of intelligence?" Most of the discussion will concern the market value of human intelligence, in order to look for useful analogies for understanding and measuring the economic value of intelligence in constructed systems.

Individuals treasure intelligence in themselves and their friends and family for a variety of reasons, most of which lead rapidly into the spiritual or metaphysical realm, or, if you prefer, into the most complex challenges of sociobiology. Either way, creating a "performance metric" for intelligence in this context seems neither feasible nor especially desirable.

On the other hand, consider the owners of a medium-sized business, who need to hire a number of employees to perform various tasks in the firm. Why should the owners pay a higher salary and go through a more difficult and expensive recruitment process to hire a more intelligent employee when they can get a less intelligent employee with the same training and experience more cheaply? To the extent we can give a quantitative answer to this question, the dollar premium a business is willing to pay for intelligence is a financial "performance metric for intelligent employees" within the context of the job at hand. Understanding how these dollar premiums arise in a variety of employment situations can give important clues on how to put a value or "metric" on the performance of intelligent machines.

There are three distinguishable ways in which a smarter employee can be worth more money to a business than a stupid one with equivalent training and experience. These are: doing what I say, doing what I want, and doing what I need.

2. Language Understanding & Capacity to Act

At the most fundamental level, "**do what I say**," an intelligent laborer can follow instructions better than a stupid laborer. Smart employees can follow instructions that are more complex, less detailed, and require less time and effort (in other words, less money) to prepare. Since they are less apt to misunderstand instructions, they require less money to be spent on supervising them than is the case for less intelligent employees with equal motivation. For constructed systems, the equivalent is an expressive command language; one that is the "natural language" for describing the task at hand, whether it resembles a spoken human language, a specialized technical language, or a graphical interface. Allied with this, of course, is the capacity to actually carry out the instructions, which some have referred to as the "body" as opposed to the "mind" of the intelligent constructed system.

3. Goal-Directedness

It is possible to view the next level, "**do what I want**," as simply an elaboration of the ability of smarter employees to follow instructions that are less detailed. However, businesses look hard for intelligent skilled craftsmen who can be told what goals to accomplish without needing to be told how to do so, and reward them with higher wages and better treatment. A major topic of discussion has been the role of **goal - directedness** in intelligent systems. In the world of human employment, a laborer (first level) is given instructions about how to do a job; the goal may be implicit in the instructions but is not an integral part of them from the laborer's point of view. A craftsman (second level), on the other hand, takes the goals provided by the employer and carries them out without further instruction. To do this, the craftsman needs experience and training, but also puts more intelligence into the work than the laborer does. [1]

Over time, a job may become more routinized, so that what originally required highly intelligent goal-seeking behavior later requires only the following of rote instructions. This can occur at either the structural level as the instructions are written down for others, or within an individual as long experience with a job eventually allows it to be done "without thinking." The equivalent to this process in the area of constructed systems would be the replacement of complex, "intelligent" processes of sophisticated search and behavior generation with stereotyped program modules or hardware gadgets, reducing the "intelligence" used by a constructed system while maintaining or even enhancing its performance.

4. Autonomy and Unpredictability

At both of the first two levels, management wants behavior of the employee to be predictable. Intelligence means autonomy in the sense that, given equivalent training and motivation, the intelligent employee does what is expected of him or her without close supervision while the stupider employee in the same job needs to be watched all the time. However, autonomy in this context is almost the opposite of creativity, spontaneity, or unpredictability; it is the stupid employee, not the smart one, who comes up with the most surprises.

It is only at the highest level, "**do what I need**," that businesses value unpredictability in their employees and consultants. Even here, there are two degrees of unpredictability. Most of the time a person or company seeks advice on matters of law, engineering, medicine, or other fields, the advice has no "information" value if the one requesting it already knew the answer; nevertheless, routine advice needs to be in line with professional standards. For example, though I do not want to be able to predict what my personal physician is going to tell me, I want it to be essentially the same as what any competent physician would say given the same knowledge about me; in other words, I want my physician's behavior to be essentially predictable by other physicians. It is only if I am suffering from an extremely serious disease, or if I am knowingly participating in a clinical experiment, that I want my physician to do something that will surprise the medical profession!

5. Information

Some of the discussion about performance metrics for intelligent systems has debated the applicability of entropy or other aspects of information theory to measuring intelligence. Fundamentally, "Information" implies informing somebody about something they didn't already know. From this point of view, an employer wants a laborer's work to provide no new information output at all, but a more intelligent laborer requires less information input that an unintelligent one. A craftsman working at the second level of "doing what I want" takes compact information about goals rather than lengthy information about procedures; the craftsman's work in sense generates "information" to the employer about the methods used, but this is information that normally is of no great interest to the employer. It is only at the highest level, that of the professional employee, that the employer is concerned about receiving information output from the employee.

		Information Input	Information Output
Laborer	Do what I say	High, procedural	Ideally none
Craftsman	Do what I want	Low, goal-oriented	Uninteresting
Professional	Do what I need	Various	Essential

6. Uncertainty

The more uncertain the job environment is, the more valuable an intelligent employee becomes. Procedural instructions about an uncertain job environment must become a complex collection of "ifs" and branches, compared to a more linear set of instructions for a job in a less uncertain environment. Businesses have to pay more for employees intelligent enough to follow such complex instructions than they do for employees whose jobs do not contain much uncertainty.

For sufficiently high levels of uncertainty in the job environment, management finds it unprofitable to prepare procedural instructions in a form that even the smartest laborer can follow. Instead, it is more economical to hire craftsmen who only need to be told the employer's goals and essentially left to implement those goals according to their own skills and

[1] *Note that my focus here is on the degree of intelligence demanded by the job, not on the intelligence possessed by the human being doing it. Job demands place only a lower bound on the worker's intelligence. Nevertheless, the more intelligence the job demands, the more the performance of an intelligent employee will overshadow that of a less intelligent one.*

intelligence. The fundamental problem with the "Chinese Room" thought experiment is that, while it might in principle be possible to prepare and index a set of stimulus-response instructions so extensive as to allow the occupant of the room to carry on a conversation in Chinese without any knowledge of the language, it is in fact such an immense task that it would be far cheaper and easier to build a machine that actually understood Chinese (and easier still to hire a human who understands Chinese to sit in the room!).

At the highest levels of uncertainty (or extreme complexity, which as Zadeh points out has many of the same effects) management can no longer be sure what goals are feasible or profitable, and so seeks expensive and potentially surprising guidance from professionals, and perhaps some day from constructed systems that produce "useful surprises" at a professional level.

7. World Models

It is very rare for an employer to ask about an employee's internal model of the world or to pay a higher salary on account of it. Laborers are paid to follow instructions intelligently in the real world, and craftsmen are paid to ply their trades intelligently in the real world. Whether or not they use an internal model of the world to do so is of no economic importance except as it is reflected, at one or more removes, in their performance.

Professionals are paid to give "useful surprises" to their employers or clients. This information (and actions informed by it) generally have to do with the real world, though at times professionals may be asked for opinions about hypothetical situations. Even then, usually it is irrelevant whether the answer comes from stored knowledge, experimentation, or the exercise of a simulation-like model in the professional expert's head. The exception is when the professional is explicitly asked to provide a model, but in that case the model is no longer an internal one, but an external analogy, flowchart, or computer simulation.

8. Self-Models and Self Awareness

Certainly, all of a firm's (human) employees have a self-model, a self-awareness, a consciousness. But only in a few "helping professions" such as psychiatry or the clergy is an abov-average endowment in this area considered an advantage to job performance. Employers value some limited facets related to self-awareness such as taking pride in one's work and being safety-conscious, but outstanding self-consciousness and self-absorption are not considered signs of outstandingly valuable intelligence by employers. Thus, with regard to constructed systems, it might be an economically important goal to build machines that "care" about doing a good job and know how to take care of themselves and those around them. But we should not insist on a robotic Mother Teresa; it would be a magnificent achievement to create a working system that was as caring and careful as a seeing-eye dog.

9. Performance Metrics

Unlike constructed systems, human employees cannot be opened up to inspect their components. Thus, employers in search of intelligent employees rely on a variety of benchmark tasks. Occasionally, they may use a benchmark task that tries to screen out the effects of knowledge to focus on pure intelligence -- examples include IQ tests and programmer aptitude tests. However, since job performance is more important than what mix of knowledge, intelligence, and other endowments it arises from, most benchmark tasks measure performance without much concern about the mix. The most common benchmark task is performance on similar jobs in the past.

Another interesting benchmark is formal education. Completing any program of study implies an ensemble of intelligence, knowledge, and skills for learning, writing, and simply sticking to a task. The education most valued by employers adds to this a body of knowledge relevant to the job. However, for complex and unpredictable environments, it may not be possible to specify in advance what body of knowledge will be required. In such a case, a broad "general education" demonstrates that a person has an advanced ability, refined by varied practice, to learn whatever is required in a new situation. With respect to constructed systems, a design team that hones and demonstrates their product's ability to learn and excel in a wide variety of problem environments, including artificial ones as well as real ones, can command a higher price for their machines than a design team that only trains their system on what is "relevant" to its expected tasks, at least from customers whose jobs are at the high end of uncertainty or complexity.

Performance metrics for intelligent systems based on board games like chess and backgammon or parlour games like the Turing test can be very useful in addressing philosophical questions about what it means to be intelligent, and technological questions about how to implement it, but they are of little direct economic interest. In particular, to pass the Turing test in a job application context, an intelligent system would have to refrain from showing any levels of ability not common among humans, and also to demand the same levels of salary and benefits as a human. What is needed, instead, is a set of benchmark tasks, probably job-specific, with one or more of the following characteristics:

- Instructions are so complicated that it is more profitable to seek an intelligent laborer system that understands them, than to seek an unintelligent "Chinese room" type system to follow the instructions without understanding.

- The environment is so complicated and uncertain that it is more profitable to seek an intelligent craftsman system that accepts exogenous goals and carries them out according to its own skills and intelligence, rather than to seek an unintelligent system that simply follows instructions.
- The situation is so fuzzy that it is more profitable to seek an intelligent professional system to determine what goals are appropriate (presumably given exogenous metagoals) and do surprising things for the benefit of the organization, rather than to seek an unintelligent system that simply and predictably carries out exogenous goals

To be useful, an intelligent constructed system must provide a better cost/benefit ratio than any combination of human being(s) and unintelligent constructed system(s). If more than one intelligent constructed system meets this test, then the one with the best cost/benefit ratio, not necessarily the smartest one, will be chosen.

10. Economics and the Vector of Intelligence

The "white paper" for the 2000 Conference on Performance Metrics for Intelligent Systems lists 25 potential coordinates for a possible Vector of Intelligence. A major challenge is to find ways to systematically quantify or otherwise specify the values of these "coordinates." Without detracting from the usefulness of methods oriented toward philosophy of mind, toward control engineering, or toward academic computer science, let me propose an economic approach to measuring each of the 25 coordinates summarized in the following table. In this economic approach, the challenge would be to estimate the derivatives of system cost/benefit ratio in a benchmark problem to "memory temporal depth," "number of objects that can be stored," ... et cetera. The second derivative is as important as the first since most or all of these coordinates are subject to diminishing or even negative returns.

Twenty-Five Potential Coordinates for the Vector of Intelligence (from the White Paper)

(a) memory temporal depth
(b) number of objects that can be stored
(c) number of levels of granularity in the system of representation
(d) the vicinity of associative links taken in account during reasoning of a situation, or
(e) the density of associative links
(f) the vicinity of the object in which the linkages are assigned and stored (associative depth)
(g) the diameter of associations ball (circle)
(h) the ability to assign the optimum depth of associations
(i) the horizon of planning at each level of resolution
(j) the horizon of extrapolation at a level of resolution
(k) the response time
(l) the size of the spatial scope of attention
(m) the depth of details taken in account during the processes of recognition at a single level of resolution
(n) the number of levels of resolution that should be taken into account during the processes of recognition
(o) the ratio between the scales of adjacent and consecutive levels of resolution
(p) the size of the scope in the most rough scale
and the minimum distinguishable unit in the most accurate (high resolution) scale
(q) an ability of problem solving intelligence to adjust its multi-scale organization to the hereditary
hierarchy of the system,
(r) dimensionality of the problem (the number of variables to be taken in account)
(s) accuracy of the variables
(t) coherence of the representation constructed upon these variables
(u) limit on the quantity of texts available for the problem solver for extracting description of the system 20
(v) frequency of sampling and the dimensionality of the vector of sampling
(w) cost-functions (cost-functionals)
(x) constraints upon all parameters
(y) cost-function of solving the problem

On the Computational Measurement of Intelligence Factors

José Hernández-Orallo

Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València, Camí de Vera s/n, 46022 València, Spain
E-mail: jorallo@dsic.upv.es.

Abstract. In this paper we develop a computational framework for the measurement of different factors or abilities usually found in intelligent behaviours. For this, we first develop a scale for measuring the complexity of an instance of a problem, depending on the descriptional complexity (Levin LT variant) of the 'explanation' of the answer to the problem. We centre on the establishment of either deductive and inductive abilities, and we show that their evaluation settings are special cases of the general framework. Some classical dependencies between them are shown and a way to separate these dependencies is developed. Finally, some variants of the previous factors and other possible ones to be taken into account are discussed. In the end, the application of these measurements for the evaluation of AI progress is discussed.

1 Introduction

Are AI systems of today more intelligent than those of 40 years ago? Probably the answer is a clear yes, at least for some of the current systems. However, another different question is 'How much more intelligent?', and, even more, in which aspects are they more intelligent?

In this paper we investigate a framework for the evaluation of such a progress in different factors, extending in a natural way the work endeavoured in [12] and [11], specific for only some inductive factors. For such an extension, the main aim should be to develop the less number of factors as possible, by proposing general factors instead of specific ones. Moreover, the framework would allow to studying their theoretical correlations, and reducing, when possible, a factor to another. This leads finally to a group of tests that can be adapted and implemented for measuring different abilities of AI systems.

First of all, we must ascertain three problems for any evaluation of the ability of solving a problem: to give a general scale of a complexity of the problem, to settle the unquestionability of the solution to the problem and to establish a way to know whether the subject has arrived to the solution.

Computational complexity scales problems according to the time different kinds of machines require to solve them in the general case by using the optimal algorithm possible. However, most problems of interest in AI are NP-complete. But, remarkably, some instances of NP-complete problems are easier than instances of polynomial problems. This assertion seems to be contradictory, since any instance has an algorithm to solve that instance in linear or even constant time (the program "if the input is x print the solution y"), so there is apparently no reason for stating that an instance can be easier than another. This has been shown to be false up to an extent, because for some problems it is better (shorter) to give a more general solution than the specific solution for an instance of the problem. This has been formalised under the notion of "instance complexity" (see e.g. [16]), which gives the shortest solution to an instance of a problem provided it does not give a contradictory solution for other instances of the same problem.

However, instance complexity is only of interest for large instances of a considerable descriptional complexity (or for sets of instances). Moreover, the difficulty of the problem is not usually related to the descriptional complexity of the solution. For instance, the descriptional complexity of the answers given by a theorem prover (an accepter) are very short, namely one bit to say 'yes' or 'no'. In the same way, the hardness of a prediction problem cannot be measured by the descriptional complexity of the element predicted, but rather by the complexity of the reason why the element has been predicted. The idea is then to measure the descriptional complexity of the 'justification' or 'explanation' of the solution. Consequently, any cognitive skill can be measured within this framework provided that problem and solution can be formalised computationally.

The paper is organised as follows. After Section 2, where some notation is introduced, Section 3 gives a general formula of the hardness of the instance of a problem, by clarifying how to generalise the concept of 'explanation' of a solution to a problem. Section 4 addresses the issue of specialising it for deductive abilities and discusses their measurement. Section 5 does the same thing for inductive abilities, but recognising that it is necessary to solve the unquestionability problem. Section 6 deals with their dependencies and the possibility of taking other factors into account. Section 7 discusses the applications of these measurements, especially for the evaluation of automated reasoning and machine learning systems. Section 8 closes the paper with the results and open problems.

2 Preliminaries

Let us choose any finite alphabet Σ composed of symbols (if not specified, $\Sigma = \{0, 1\}$). A string or object is any element from Σ^*, with $\circ$ being the composition operator, usually omitted. By $\langle a, b \rangle$ we denote a standard recursive bijective encoding of a and b, such that there is a one-to-one correspondence between $\langle a, b \rangle$ and each pair (a, b). Note that this usually takes more bits than $a \circ b$. The empty string is denoted by ε. The term $l(x)$ denotes the length or size of x in bits and $\log n$ will always denote the binary logarithm of n.

The complexity of an object can be measured in many ways, one of them being its degree of randomness [14], which turns out to be equal to the shortest description of it. Descriptional Complexity, Algorithmic Complexity or Kolmogorov Complexity was independently introduced by Solomonoff, Kolmogorov and Chaitin to formalise this idea, and it has been gradually recognised as a key issue in statistics, computer science, AI and cognitive science [16][6].

The Kolmogorov Complexity of an object, defined as the shortest description for it, usually denoted by C (plain complexity) or K (prefix-free complexity) turns out to be not computable in general, due to the halting problem. One solution for this is to incorporate time in the definition of Kolmogorov Complexity. The most appropriate way to weight space and time execution of a program, the formula $LT_\beta(p_x) = l(p_x) + \log \tau_\beta(p_x)$, where τ is the number of steps the machine β has taken until x is printed by p_y, was introduced by Levin in the seventies (see e.g. [15]). Intuitively, every algorithm must invest some effort either in time or demanding/essaying new information, in a relation which approximates the function LT. The corresponding complexity, denoted by Kt (see e.g. [16]) is a very practical alternative to K.

3 Problem Complexity by Its Explanation Complexity

Consider a problem instance π as a tuple $\langle S, C, I, A, \phi \rangle$ where S is the context or working system where the problem can be established, C is a Boolean function which represents a (syntactical) validity criterion, I is the presentation of the instance, A_i is the answer and ϕ is a (semantical) verifier[1]. The general problem is denoted by $\pi(\cdot)$ as the tuple $\langle S, C, \phi \rangle$.

We say that E is an explanation for the problem instance π iff E is valid, i.e. $C(\langle S, I, E \rangle) = true$, and E is a means to obtain the solution, i.e., $\phi(\langle S, I, E \rangle) = A_i$.

From here, it is easy to adapt the definition of Kt to measure the hardness of a problem. Namely, the hardness of a problem instance $\pi\langle S, C, I, A, \phi \rangle$ is then defined as:

$$H(\pi) = \min\{LT(E|\langle S, C, I \rangle) : E \text{ is an explanation for } \pi\} \tag{1}$$

[1] Both C and ϕ could be joined in one function. We have preferred to separate them, because later it will be useful to distinguish between both parts of a correct solution, in order to establish purer factors.

For instance, the hardness of a search problem is usually estimated by the size of the search space. If the search problem is complex, it is necessary to say which branches have been selected in order to arrive to the solution, or either a long time is necessary to explore (and make backtracking) to the misleading ones. It is the function LT which finds a compromise between the information which is needed to guide the search and the logarithm of the time that is also needed to essay all the branches. On the other hand, if the search problem is linear (one possible branch), it is very easier to describe the problem (just follow the rules in the only possible way). However, for very long derivations, the inclusion of time can make hardness high too.

For the evaluation of a subject's ability of solving a kind of problem $\pi(\cdot)$ it is necessary to generate a set of instances of that problem of different hardness. In order to scale the instances more properly, we introduce the concept of k-solvability. An instance of a problem $\pi = \langle S, C, I, A, \phi \rangle$ is *k-solvable* iff k is the least positive integer number such that:

$$H(\pi) \leq k \cdot \log l(I) \tag{2}$$

The use of $\log l(I)$ is justified by the fact that, once the general problem is known, each instance must be 'read' an this takes at least $l(I)$ steps.

Once given a general scale of a complexity of the problem, it is then easy to make a test from the previous definition, provided that the unquestionability of the solution to the problem is clear. Unquestionability can only be addressed depending on the kind of problem (we will see this for deductive abilities and especially for inductive abilities in the following sections). Finally, there is no way to know whether the subject has arrived to the solution if the explanation is not given (and usually the explanation is difficult to check or the subject may not be able to express the explanation in a comprehensible form). For instance, the subject may have given the right solution but maybe due to wrong derivations. Fortunately, in the case of multiple solutions, this situation will be discardable in the global reckoning of the test. In the case of few solutions, such as 'yes'/'no', it is then necessary to penalise the errors by using some formula that takes into account the possibility of guessing the right answer 'by error'.

Another question is the time limit for making the test. This would highly depend on the factor to be measured, and whether there is a special interest on evaluating the ability to solve a given problem or the ability to solve it quickly. The selection of the time limit and the evaluation of the score according to it could be very interesting for evaluating resource-bounded rational systems.

Finally, we have not considered the possibility of multiple correct explanations for the same solution, which would suggest a modification of (1). Consider the situation of the best explanation with $LT = n$, but several other explanations of $LT = n + 1$. Intuitively, the existence of these other explanations also affects the easiness of the solution. However, this is very difficult to evaluate in practice because there are always infinite slight variations of the best explanation (void steps, redundancies, etc.), so the previous situation is extremely frequent (if not inevitable). It is then assumed that for every k:

$$\begin{array}{ll} card\{ \ E : LT(E) = k \text{ and } C(\langle S, I, E \rangle) = true \text{ and } \phi(\langle S, I, E \rangle) = A_i \ \} << \\ card\{ \qquad E : LT(E) = k \text{ and } C(\langle S, I, E \rangle) = true \qquad \} \end{array} \tag{3}$$

In other words, we assume that the proportion of valid and correct explanations wrt. valid explanations is very small.

Once a general framework is established, let us study which deductive and inductive abilities are feasible and interesting to be measured within it.

4 Deductive Abilities

Apparently, deductive abilities are much easier to measure, because there is no possible subjectivity in the correct answer; given the premises and the way to operate with them, only one answer is possible.

An instance of a deductive problem $\pi = \langle S, C, I, A, \phi \rangle$ can be defined in terms of the previous framework in the following way: S corresponds to the set of axioms or axiomatic system, C is a Boolean function which says what is a valid application of the axioms, I is the instance of the deductive problem, A_i the answer and ϕ is a verifier, i.e., $\phi(\langle S, I, E \rangle) = A_i$, in this case, a verifier that checks whether A_i is a result of applying a solution to I in S.

In this case the explanation E is represented by a *proof* in S stating that A_i is a the result of I or, in other words, a derivation from I to A_i.

Example: Consider for instance an accepter that tells whether a proposition is a theorem or not. Let S be the axioms of arithmetic. Let C a function that tells that a derivation is valid according to the rules of application of the axioms, and let I be the instance "Is Fermat's famous conjecture true?" (recently a theorem). Which is the hardness of the solution A = 'yes'? The descriptional complexity of A (which is just yes) would say that the instance is very easy, however its hardness given by H turns out to be the LT of the proof with less LT. Consider instead the instance "solve 2+3" which, also with a low complexity of $A = 5$, turns out to be simple, because the derivation is describable easily and shortly from $\langle S, C, I \rangle$. In general, any calculation is shortly describable, so its hardness will depend solely on its temporal cost.

According to this example, we can distinguish some classical deductive problems that can be measured. In particular, the following factors are distinguished:

- Calculus Ability: in this special case, C only allows a specific and deterministic application of the rules or axioms of S. In this case the search space is linear. As it has been said before, its complexity is exclusively given by the logarithm of the time which is needed from the input I to the output A_i. This ability is not of much interest to be measured nowadays, since it is better done by computers than humans, and it would finally measure the computational power of the subject / machine.
- Derivational Ability: in this case, C only allows a varied application of the rules or axioms of S. Consequently, the search space is open. The complexity is then given by a compromise between the logarithm of the time which is needed to know that a branch leads to no solution, and some information that may say which branches to take (and which ones not to take).
- Accepter Ability (proving ability): It is a special case of the previous ability, with the special feature that I can only be 'yes' or 'no'. Theoretically, there is no reason for expecting that a subject has a different result in this problem that in the previous one.

The way to implement a concrete test for the previous ability is not complicated. For calculus ability, it is just necessary to generate some derivations. Their length will determine the time which is needed to follow them. On the contrary, for the other two abilities, it is necessary to generate a possible derivation, and look that there are no shorter equivalent derivations. This, in general, will be extremely costly, growing exponentially according to the value of k-solvability. Fortunately, there is no need for efficiency here. A hard test can be generated during days, even weeks, and then passed to several subjects.

5 Inductive Abilities

A sequential inductive problem $\pi = \langle S, C, I, A, \phi \rangle$ can also be defined in terms of the previous framework in the following way: S corresponds to the background knowledge, I is a sequential evidence (with $l(I) = n$), C is a Boolean function which represents the hypothesis selection criterion (e.g. simplicity), A_i is the prediction of the $(n+1)$th element of the sequence and ϕ is a verifier, i.e., $\phi(\langle S, I, E \rangle) = A_i$, in this case, a verifier that checks whether A_i is the $(n+1)$th element given by the hypothesis with the background knowledge S and also checks whether both cover I.

In this case the explanation E is represented by a 'hypothesis' wrt. S that affirms that A_i is 'what follows' I or, in other words, a prediction from I.

Example: Consider for instance a prediction problem. Let S be a background knowledge, containing, among other things, the order of the Latin alphabet. Let C a function that tells that a hypothesis is good according to a selection criterion, and let I the instance "aaabbbcccdddeeefffgggh". Which is the hardness of the solution A_i = 'h'? The descriptional complexity (in LT terms) of the hypothesis is again what is taken into account.

The main question of evaluation of induction is that of inquestionability. Even if the selection criterion is given, two plausible explanations may differ slightly, and the selection criterion would give that one is slightly better than the other, but this would depend highly on the descriptional mechanism used. In [12] and [11] this difficult problem is addressed, according to a comprehensive criterion, a variant of the simplicity criterion based on Kolmogorov Complexity in the style of Solomonoff [19], but ensuring that the data is covered comprehensively, i.e. without exceptions. Accordingly, the *simplest explanatory description*, denoted by $SED(x|y)$, is defined in [11] as the simplest (in LT terms) description which is comprehensive wrt. the data x given the background knowledge y. To ensure unquestionability, the examples are selected such that there are no alternative descriptions of similar complexity that give a different description. Finally, there is a small possibility that a good prediction is given by a 'wrong' explanation. This probability may be neglected in the tests or corrected by a penalising factor in the score of wrong results.

From here, partially independent factors can be measured by using extensions of the previous framework. For instance, inductive abilities, such as sequential prediction ability, knowledge applicability, contextualisation and knowledge construction ability can be measured in the following way:

- Sequential Prediction Ability: several unquestionable sequences of different k-solvability are generated. A test for this ability has been generated in [12] and passed to humans, jointly with a typical psychometrical test of intelligence. The correlation showed that this is one of the fundamental factors of intelligence, although more experimentation is to be done.
- Inductive Knowledge Applicability (or 'crystallized intelligence'): a background knowledge B and a set of unquestionable (with or without B, denoted by $H(x_i|B)$ and $H(x_i)$ respectively) sequences x_i are provided such that $H(x_i|B) = H(x_i) - u$ but still $SED(x_i|B) = SED(x_i)$. The difference of performance between cases with B and without B is recorded. This test would actually measure the application of the background knowledge depending on two parameters: the complexity of B and the usefulness of B, measured by u.
- Inductive Contextualisation: it is measured similarly as knowledge applicability but supplying different contexts $B_1, B_2, ..., B_T$ with different sequences $x_{i,t}$ such that $H(x_{i,t}|B_t) = H(x_{i,t}) - u$. This multiplicity of background knowledge (a new parameter T) distinguishes this factor from the previous one.
- Inductive Knowledge Construction (or learning from precedents): a set of sequences x_i is provided such that there exists a common knowledge or context B and a constant u such that for $H(x_i|B) \leq H(x_i) - u$. A significant increase of performance must take place between the first sequence and the later sequences. The parameters are the same as the first case, the complexity of B and the constant u.

It is obvious that these four factors should correlate, especially with the first one, which constitutes a necessary condition for having a minimal score in the other factors.

6 Dependencies and Other Factors

Although there is a common (but argueable) view of induction and deduction as inverse processes, they are not inverse in the way they use computational resources. In fact, any inductive process requires deduction to check the hypotheses, thus, obviously, inductive ability is influenced by deductive ability. This has been usually recognised by IQ tests, where deductive and inductive abilities usually correlate. Due to this fact, inductive factors usually are the main part of intelligence tests, because deductive abilities are implicitly evaluated.

However, if we are looking for 'pure' factors the question is whether there is a way to separate this deductive 'contamination' in inductive factors.

The idea is to provide 'external' deductive abilities when measuring inductive factors, in order to 'discount' the deductive effort than otherwise should be done. For this, given a problem $\pi = \langle S, C, \phi \rangle$ it is only necessary to provide an 'oracle' which computes ϕ in constant time. The subject must only guess models (hypotheses) and check them in the oracle, by providing the hypothesis to it and comparing the results with the evidence I. This would measure the 'creative' part of induction. In the following, let us denote by 'purely' inductive the corresponding factors to those highlighted in the previous section which result from providing the oracle.

This resembles a 'trial and error' problem considering reality acting as the oracle. The issue is how to implement this in a feasible way, especially for evaluating complex agents or even human beings. The best way, in our opinion, is the construction of a 'virtual' world where the subject to be evaluated can interact and essay its hypotheses with no effort.

In a similar way as the oracle for ϕ, some difference could be estimated if the syntactical machine C is (also) given. Although this would not be much representative for deduction, for induction it would discount the ability of working with the selection criterion, which is an important trait of induction.

Nonetheless, deductive ability is also influenced by inductive ability as long as the problems become harder. Some lemmata or rules can be generated by an intelligent subject in order to help to shorten the proof from the premises to the conclusion. This may explain why artificial problem solvers without inductive abilities have not been able to solve complex problems, and this is especially clear in Automatic Theorem Proving. Consequently, recent systems are beginning to use ML techniques for improving performance. Background knowledge could also be examined in deduction, provided S includes the axioms but also some useful properties. This finally gives similar factors as those given for induction:

- Deductive Knowledge Applicability: how lemmata or properties are used for a deductive problem.
- Deductive Contextualisation: the ability of using different contexts for different problems.
- Deductive Knowledge Construction: this will measure the increase of performance between first instances and last ones.

Finally, we have given a measurement for sequential induction, and it seems interesting to evaluate non-sequential induction as well, where an unordered set of elements is given as evidence from an unknown function that maps whether an element belongs to a set. In this case, the test could give some possible values which might be members of the set, although only one of them is really in it. Solomonoff formalised deterministic (sequential) prediction [19] and recently, has formalised non-sequential prediction [21]. This problem is similar to the inductive problem of learning a Boolean classifier and can be extended to the case of a general classifier. To eliminate the deductive contamination of the measurement of non-sequential induction, the 'oracle' ϕ should be a classifier, telling, given a hypothesis, to which class the element belongs. The essay of an 'oracle' that accepts several elements at a time should be considered as well.

Once the basic deductive and inductive factors have been recognised, the question is whether there are many other factors which are relevant to be measured. For instance, memory or 'memoisation ability' is a factor that is knowledge-independent and it can be easily measured. However, this factor is not very interesting for AI nowadays.

Other factors, such as analogical and abductive abilities can be shown to be closely connected to inductive and deductive abilities both theoretically and experimentally. A first approach for measuring them has been attempted in [12], and the test applied to human beings has shown the correlation with inductive abilities.

However, not every factor is meaningful. Factors like "playing chess well" are much too specific to be robust to the subject's background knowledge. However, it cannot be discarded that some game-playing factor would measure competitivity and interactivity abilities aside from deductive and inductive abilities.

Finally, we have considered individual tests which measure one factor. For measuring several factors at a time, the exercises should be given one by one and, after each guess, the subject should be given the correct answer (rewards and penalties can be used instead). This has two advantages: there is no need for the subject to understand natural language (or any language) to order to be explained the purpose of the test, and there is no need to tell which factor or purpose is to be measured in each part of the test. There is also one disadvantage, deductive problems should be posed in terms of 'learn to solve', and this may devirtualise them.

7 Applications

Modern AI systems are much more functional than systems from the sixties or the seventies. They solve problems in an automated way that before required human intervention. However, these complex problems are solved because a methodical solution is found by the system's designers, not because most current systems are more intelligent than preceding ones. Fortunately, the initial aim of being more general is still represented by some subfields of AI: automated reasoning and machine learning.

Automated reasoning (more properly called Automatic Theorem Proving) is addressing more complex problems by the use of inductive techniques, while maintaing their general deductive techniques. These systems, in fact, have been used as the 'rational core' of many systems: knowledge-based systems, expert systems, deductive databases, ... But, remarkably, the evaluation of the growth of automated reasoning has not been established from the success of these applications but from the increasingly better results on libraries of problems, such as the TPTP library [22]. However, there is no theoretical measurement about the complexity of the problems which compose these libraries. Instead, some approximations, such as the number of clauses, use of some lemmatas, etc., have been used. Following the approach presented in this paper it would be interesting to give a value of $k - solvability$ of each of the instances of these libraries.

In a similar way, machine learning has recently taken a more experimental character and systems are evaluated wrt. sets of problems. Except from general problems (classes), where their complexity (or learnability) has been established, there is no formal framework for giving a scale for concrete instances.

In this new and beneficial interest in measurement, Bien et al. [1] have defined a 'Machine Intelligence Quotient' (MIQ), or, more precisely, two MIQs, from ontological and phenomenological (comparative) views. Any comparison needs a reference, and the only reference of intelligence is, for the moment, the human being. This makes the approach very anthropocentric, like the Turing Test. The ontological approach, however, is not based on computational principles but on a series of characteristics of intelligence that are defined on linguistical terms rather than computational/mathematical ones, such as long-term learning, adaptation, recognition, optimization, etc. Moreover, the evaluation is generally measured on performance on some specific problem, contrary to the claim that "it is time to begin to distinguish between general, intelligent programs and the special performance systems" [18]. Although this can be very appropriate for specific systems where functionality is clear, in general this would not allow for the comparison of intelligence skills of different systems devised for quite different goals. How to define general and absolute characteristics of intelligence computationally is more difficult and new problems present themselves, but the progress in the 'intelligence' of AI systems can only be measured in this way.

8 Conclusions and Future Work

Among the problems for making these measurement reliable there is the selection of a reference machine. The evaluation of abilities with instances is dangerous because it depends on constants. Since there is no apparent preference for any descriptional mechanism, we plan to adapt these notions for logic programming, because it is a paradigm that has been used both for automated deduction and machine learning (ILP) as well as other uses (abduction, theory revision, ...), and, in our opinion, is not biased.

For the moment, the framework which has been presented allow for the measurement of different factors and clarifies the distinction between evolutionary-acquired knowledge, life-acquired-knowledge and 'liquid intelligence' (or individual adaptability). Several tests for different subfields of AI could be devised following this paradigm, and the increasing scores for solving more and more complex (k-solvable) problems may be a way to know how much intelligent AI systems are wrt. previous generations systems.

References

1. Bien, Z., Kim Y. T. and Yang, S. H., 1998, "How to Measure the Machine Intelligence Quotient (MIQ): Two Methods and Applications", *World Automation Congress (WAC)*, TSI Press, Albuquerque, NM.
2. Blum L. and Blum M., 1975, "Towards a Mathematical Theory of Inductive Inference". *Inf. and Control*, 28:125–155.
3. Bradford P. G. and Wollowski, M., 1995, "A Formalization of the Turing Test (The Turing Test as an Interactive Proof System)". *SIGART Bulletin*, **6**(4), p. 10.
4. Chaitin, G. J., 1982, "Gödel's Theorem and Information". *Int. J. Theo. Phys.*, **21**, 941-54.
5. Eysenck, H. J., 1979, *The Structure and Measurement of Intelligence*, Springer-Verlag.
6. Gammerman, A. and Vovk, V. (eds.), 1999, Special Issue on Kolmogorov Complexity, *The Computer Journal*, **42**(4).
7. Gold, E. M., 1967, "Language Identification in the Limit", *Inform & Control*, **10**, 447-474.
8. Harman, G., 1965, "The inference to the best explanation", *Philos. Review*, **74**, 88-95.
9. Herken, R., 1994, *The universal Turing machine: a half-century survey*, Oxford Univ. Press, 1988, 2nd Ed., 1994.
10. Hernándcz-Orallo, J., 2000, "Computational Gain and Inference", *Collegium Logicum*, **4**, Springer, in press.
11. Hernández-Orallo, J., 2000, "Beyond the Turing Test", to appear in *Journal of Logic, Language and Information*, to appear in Vol. 9 no. 4.
12. Hernández-Orallo, J. and Minaya-Collado, N., 1998, "A Formal Definition of Intelligence Based on an Intensional Variant of Kolmogorov Complexity" In *Proc. of the Intl. Symp. of Engin. of Intelligent Systems* (EIS'98), ICSC Press, 146–163. 1998.
13. Johnson, W. L., 1992, "Needed: A New Test of Intelligence", *SIGART Bulletin*, **3**(4), 7–9.
14. Kolmogorov, A. N., 1965, "Three Approaches to the Quantitative Definition of Information", *Problems Inform. Transmission*, **1**(1):1-7.
15. Levin, L. A., 1973, "Universal search problems", *Problems Inform. Transm.*, **9**, 265-6.
16. Li, M. and Vitányi, P., 1997, *An Introduction to Kolmogorov Complexity and its Applications*, 2nd Ed., Springer-Verlag.
17. Neisser, U., Boodoo, G., Bouchard, T. J., Boykin, A. W., Brody, N., Ceci, S. J. Halpem, D. F., Lochlin, J. C., Perloff, R., Sternberg, R. J. and Urbina, S., 1996, "Intelligence: Knowns and Unknowns", *American Psychologist*, **51**, 77–101.
18. N. J. Nilsson, Eye on the Prize. *AI Magazine*, July 1995.
19. Solomonoff, R. J., 1964, "A formal theory of inductive inference", *Inf. Control*, **7**, 1-22, March, 224–254, June.
20. Solomonoff, R. J., 1978, "Complexity-based induction sytems: comparisons and convergence theorems", *IEEE Trans. Inform. Theory*, IT-**24**, 422–438.
21. Solomonoff, R. J., 1999, "Two Kinds of Probabilistic Induction", in the 'Special Issue on Kolmogorov Complexity', *The Computer Journal*, **42**(4), 256–259.
22. Suttner, C. B. and Sutcliffe, G., 1998, "The TPTP Problem Library: CNF Release v1.2.1", *Journal of Automated Reasoning*, **21**(2), 177–203.
23. Turing, A. M., 1936, "On computable numbers with an application to the Entscheidungsproblem", *Proc. London Math. Soc.*, series 2, **42**, 230–65, 1936. Cor., Ibid, **43**, 544–6, 1937.
24. Turing, A. M., 1950, "Computing Machinery and Intelligence", *Mind*, **59**, 433–460.

ON DEFINITION OF TASK ORIENTED SYSTEM INTELLIGENCE

Michel Cotsaftis
LTME/ECE 53 rue de Grenelle 75007 Paris France
fax:33-(0)1-42-22-59-02; email:mcot@ece.fr

ABSTRACT

With development of system complexity and performances, it is important to evaluate its ability to perform tasks, especially in the case of opposing outer effects. This amounts to affect "intelligence" coefficient to the system, which basically requires to transfer usual geometric space calculations to more global and qualitative task space, the only one where this coefficient can have a meaning irrespective of system structure. The problem is discussed here by defining the useful information by its analytical expression explicit in terms of system elements. By application to the class of deformable Lagrangian systems, adapted controlled structure is constructed. Intelligence measured by minimization of a distance between demand and result mainly appears as a compromise between information ball and robustness ball reduction for fixed system complexity.

Keywords : *System complexity, Functional Asymptotic Control, Useful Information and Entropy, Intelligence, Task Space Control.*

1-INTRODUCTION

As technical systems required for real life task accomplishment are becoming very complex both in their (hard ware) physical realization and in the related (software) organization of their command-control structure, an emerging question is in the possible existence of a limit in improving these systems. Supposing everything can be continuously extended on hardware side, a direct consequence on soft side is the research of a quantitative way to scale system capability, ie in short to measure their "intelligence"[1]. One should first make sure that the question has a well defined meaning as for human the definition of intelligence is multiform and depends on the emphasized "qualities" in the tests. Also, a difficulty is the domain on which this "intelligence" is applied, as there exists different kinds of human "intelligence" ranging from high abstraction to very applied domains. To avoid these problems the angle of approach will be modified and, as a system is generally designed for accomplishing a prescribed set of tasks, its "intelligence" compared to another system will be evaluated in terms of its "efficiency" to collect the relevant information for these tasks and to use it in its accomplishment. A companion question is system adaptation to different or even adverse working conditions, which also amounts to evaluate the size of robustness ball corresponding to the selected tasks. A difficulty however reappears with the word "selected" as concerns "who" is chosing the tasks, and this stresses the huge difference between dedicated and self-deciding system structures. In first case, "intelligence" measurement is limited to evaluation of simple faithfullness in design and organization, and to robustness to parameter change, whereas in second one, a new dimension in system evaluation capability is added, showing that the problem cannot be handled in an universal and unique framework.
Another strong restriction is coming from hardware. Example of lightweight robot arms[2] shows that for high enough power there exists a breakpoint where internal material structure generates excitation of internal deformation modes impairing initially researched performances. One may speculate that this could be cured by adequate controller design using vision system, most adapted to detect working environment and to give more flexibility to adapt to task change. As mounting vision sensor on robot arm is no longer possible with deformations, exterior more rigid fixture should be used. If environment is then correctly observed, robot arm vibrations still remain, forbidding fast enough approach to target. So including robot end effector in visual observation may appear as a natural solution, but the reality is that this is not possible as actuator frequency range is significantly smaller than typical perturbation frequency range. Active control robustification, a constant trend in control research over the last decades, becomes inefficient beyond some today crossed limit because of the inavoidable spillover from low frequency actuator range to high frequency internal system range which severely limits the performances. This internal contradiction (more controlled active power for nominally better performance

leading to secondary internal phenomena downgrading more this performance) also makes the "intelligence" assessment somewhat questionnable in the present context, and bounds even more the domain where the problem has a well defined meaning. Interpretation is that usefulness of collected information from sensors is strongly system depending, including human operator, raising the problem of its adequate selection for a given system and a prescribed task.

Escaping from these difficulties is however possible by observing that this limitation comes from unability of computation-control system to reconstitute, as it classically does, actuator command from trajectory observation for its efficient control. Two different elements are implied in this statement. One is the impossibility past some level of complexity to distinguish between two close enough trajectories. Even with perfect end effector location in time and space, decomposition of this observation on base representation functions becomes unrealizable when flexion and torsion effects are mixing up in a very complicated motion. Control action becomes inefficient if one-to-one relation between control and trajectory is no longer maintained. Even if it were maintained, the power would have to be delivered, owing to speed and torque requirements, in a too high frequency range for present actuators, and this would be technically non realizable. The second element is also of fundamental nature in that there is no direct action on deformation modes from actuators, as they are receiving their power input from rigid motion mode, leading to a mismatch between internal natural power cascade and external one imposed by feedback loop with usually spillover effect impairing again system performance.

As there is inadequation between basic physics understanding and new bifurcated situation, classical point of view should be changed. With trajectory non distinguishability the base ingredient for trajectory control, ie its time dependence in usual state space representation, should be abandonned. Only trajectories as a whole have now a meaning, and global enough information is relevant. Reducing the complete non controllable system dynamics to smaller initially driving rigid ones, time dependent system trajectory is embedded into a selected class by application of fixed point theorem. The resulting control, explicitely expressed in terms of global system quantities, still gives asymptotic stability toward desired trajectory, and exhibits the interesting property to be at its level naturally organized toward task orientation. So in progressing toward higher quality performances with higher designed and more complex systems, use of better components is not sufficient and control structure has also to fit with system properties, implying mainly application of *subsidiarity principle* guaranteeing minimization of internal information flux. This restores adjustment of system hardware structure to possible task assignment, as it gives again the system the way to have appropriate internal information exchange compatible with power flux. Resulting internal coherence thus appears as an extremely important element in the possibility of measuring system "intelligence".

To illustrate the previous concepts developed at system level, *useful* information is defined in next paragraph and task oriented control for general Lagrangian system dynamical equations is considered. Application to actuated one-link robot arm with flexion and torsion deformations carrying of-center massive object is discussed with Euler-Bernouilli approximation. When compared to usual control based on vision system which in present case cannot insure trajectory stability, "local" deformation effects are internally taken care of by proposed control. As much lower information flux circulation is implied, vision system is freed for higher level task of driving the approach to desired target, and for much more modest computing requirement. In this sense, actual system may appear as more "intelligent".

2-SYSTEM REPRESENTATION AND USEFUL INFORMATION

For global system improvement, system parts have themselves to be improved in their various components. Basically three hardware parts always exist in a system, 1)- a mechanical-physical part, 2)- a sensor-computing part, and 3)- a power-actuation part, see Fig.1. There also exists a fourth software control law part, which should enable the system to correctly perform in targetted range within its new physical conditions, manifested by the creation of a (possibly infinite) number of internal modes, thus increasing its number of degrees of freedom, and making previous classical controls inadapted. The control based on the new physical conditions theoretically exists[2] and still makes system trajectories asymptotically stable, ie it guarantees again tracking performance requirements.

Due to larger excited frequency band when mode number increases, the problem now rests upon 1)- sensing and treating this new added information, and 2)- generating the corresponding power inputs as needed for increasing system performances. The first point belongs to sensor-computing part, and is handled within existing technology covering a large frequency band with a wide set of technical solutions and corresponding to broad range of accuracy. For the second point, despite the large size domain ($[10^{-1}m, 10^{1}m]$) without going into more specific microsystems, there still exists a frequency gap between classical actuators low frequency domain ($[0, 30Hz]$), and high frequency domain corresponding to "smart" material systems ($[3.10^2 Hz, 3.10^3 Hz]$). Any new information is directly usable only if it belongs to the intersection of both sensors and actuators frequency ranges. A very striking case is vision sensor giving an over-detailed amount of informations not directly useful for system control improvement. Consequently to give the system adapted capability, the problem is not in getting more information as believable from the increase of system internal degrees of freedom, but on the contrary to reduce the extra-information from state space in frequency range outside actuator's one, and in order to maintain robust asymptotic stabilization by adapted control within the uncertainty ball corresponding to the unpreciseness produced by this reduction. As shown on Fig.1, it is after collection of rough information from sensors that there should exist a reduction process to filter the only relevant information needed for reaching system targets. This leads to the definition of *useful information* determined from task orientation rather than lower level unexploitable trajectory orientation. It is based on observation that occurrence of events rests upon removal of a double uncertainty : the usual quantitative one related to occurence probability and the qualitative one related to event utility for goal accomplishment. So events may have same probability but very different utility, and this explains why some extra informations on top of existing ones have no impact on reaching the goal. In present case, it can be verified that, calling u_j and p_j the utility and the probability of event E_j, and $I(u_j, p_j)$ its associated information called useful information, the relation

$$I(u, p_1 p_2) = I(u, p_1) + I(u, p_2) \qquad (1)$$

holds for event $E_1 \cup E_2$ with same utility u. On the other hand, there is strict proportionality between utility and corresponding information, so

$$I(\lambda u, p) = \lambda I(u, p) \qquad (2)$$

With eqns(1,2), there results that useful information is given by

$$I(u, p) = -ku.\log p \qquad (3)$$

where k is Boltzmann constant. Usual entropy calculation is thus obtained by presupposing that all events have same utility for goal accomplishment, which is certainly true in Thermodynamics where all molecules are totally interchangeable and thus indistinguishable. As a consequence it is well known that only the invariant corresponding to this equivalence class, here the energy (or the temperature), allows to separate thermodynamical systems. Similarly internal system deformations (flexion and torsion) are undistinguishable events as they are layered on invariant surfaces determined by the value of bending moment M at link's origin[3]. So using their observation to improve system dynamical control is not possible, in the same way as observing individual molecule motion in a gas does not improve its global control. As a result, raw sensor information has to be filtered so that only useful information for desired goal is selected. This is precisely the remarkable capability of living systems to have evolved their internal structure so that this property is harmoniously embedded at each level of organisation corresponding to each level of development. In this sense they are remarkably intelligent. An important element here is that the process has been subsidiased into the hardware structure in order to free the upper levels.

3-LAGRANGIAN EQUATIONS FOR DEFORMABLE SYSTEM

To proceed, advantage will be taken of the general lagrangian form of deformable system in order to exhibit directly on system equations the features discussed above concerning information reduction. First there is a cascade effect of exterior forces onto rigid dynamics feeding itself deformation modes, allowing reduction of complete initial (infinite dimensional) system to (finite dimensional) "core" rigid system, see Fig.2. Then, and as long as "natural" boundary conditions are considered for the system, only these intrinsic elements will be really needed to control system dynamics. By "natural" are meant boundary conditions constructed with the remaining terms coming from the various integrations by part needed to transform system action variation into Lagrange equations. More specifically, with Lagrangian density

$$\mathcal{L}_T = \mathcal{L}_T \left(q_j(t), \frac{dq_j(t)}{dt}, u_k(x,t), \frac{\partial u_k(x,t)}{\partial t} \frac{\partial^m u_k(x,t)}{\partial x^m}, \right. \\ \left. u_k(S_1,t), \frac{\partial u_k(S_1,t)}{\partial t}, \frac{\partial^m u_k(S_1,t)}{\partial x^m}, x, t \right) \qquad (4)$$

depending on both discrete (rigid) variables $q_j(t)$ and field (deformation) variables $u_j(x,t)$ up to their pth space derivatives, as well as their values on a part (S_1) of total system boundaries ($S = S_1 \times S_2$) of the space domain $D(x)$ in the additive form

$$\mathcal{L}_T = \frac{1}{V(x)} L_R \left(q_j, \frac{dq_j}{dt}, t \right) + \mathcal{L}_D \qquad (5)$$

of a rigid variable part L_R and a deformable one $\mathcal{L}_D$, and where the arguments in the second part are the same as in eqn(1). The variation of the action

$$\mathcal{I} = \int_{t_0}^{t_f} \int_{D(x)} \mathcal{L}_T dx dt \qquad (6)$$

inside the space domain $D(x)$ and over the time interval $[t_0, t_f]$ can finally be splitted into two parts, one under the integral sign and another one expressed at the boundary (S) of $D(x)$ and at the limits of the time interval (if there are "transversality conditions"), and resulting from integrations by part. Writting that system equations are deduced from the action $\mathcal{I}$ by a variational principle implies two elements :

- 1 - the Lagrange equations

$$\frac{\partial \int \mathcal{L}_T}{\partial q_j} - \frac{d}{dt}\frac{\partial \int \mathcal{L}_T}{\partial \dot{q}_j} = F_j + U_j$$

$$\frac{\partial \mathcal{L}_T}{\partial u} - \partial_\mu \frac{\partial \mathcal{L}_T}{\partial u_\mu} + \partial_{\mu\nu}\frac{\partial \mathcal{L}_T}{\partial u_{\mu\nu}} - \cdots - \frac{d}{dt}\frac{\partial \mathcal{L}_T}{\partial \dot{u}} = 0 \qquad (7)$$

are satisfied inside the space-time domain, with U_j the control acting onto the system,

- 2 - the remaining boundary terms resulting from integration by parts are equated to the work done by exterior force terms onto the system, ie.

$$\begin{aligned} &\left(n_\mu . \left[\frac{D\mathcal{L}_T}{Du_\mu} + \cdots \right]_{S_j} + \frac{\nabla \mathcal{L}_T}{\nabla u(S_j)}\delta_{j1} \right) .\delta u_{S_j} = 0 \\ &\left(n_\nu . \left[\frac{\partial\mathcal{L}_T}{\partial u_{\mu\nu}} - \cdots \right]_{S_j} + \frac{\nabla \mathcal{L}_T}{\nabla u_\mu(S_j)}\delta_{j1} \right) .\delta u_{\mu S_j} = 0 \end{aligned} \qquad (8)$$

with

$$\nabla \mathcal{L}_T / \nabla Z = \partial \mathcal{L}_T / \partial Z(S_j) - d/dt[\partial \mathcal{L}_T / \partial \dot{Z}(S_j)]$$

$$D\mathcal{L}_T / Du_\mu = \partial \mathcal{L}_T / \partial u_\mu - \partial_\nu \partial \mathcal{L}_T / \partial u_{\mu\nu}$$

and transversality conditions if any are satisfied. Boundary conditions are called "natural" when they are constructed from these quantities, and not from different ones.

For a 1-link system, the lagrangian writes in partitioned form

$$\begin{aligned} \mathcal{L} =& L_r(q_j(t), \dot{q}_j) + \int \mathcal{L}_d(q_j, \dot{q}_j, q(x,t), \dot{q}, q_\mu, q_{\mu\nu}) \\ &+ \mathcal{K}_S(q_j, \dot{q}_j, q_S, \dot{q}_S, q_{\mu S}, \dot{q}_{\mu S}) \end{aligned} \qquad (9)$$

with rigid part

$$L_r = J_a \left(\frac{d\theta}{dt}\right)^2 + J_m \left(\frac{d\theta_m}{dt}\right)^2 + K_m(\theta - \theta_m)^2 \qquad (10)$$

in terms of rigid articular and actuator variables $q_1 = \theta$, $q_2 = \theta_m$, deformation part

$$\begin{aligned} \mathcal{L}_d =& \rho A \left(x\frac{d\theta}{dt} + \frac{\partial u(t,x)}{\partial t} \right)^2 + \rho K^2 \left(\frac{\partial \gamma(t,x)}{\partial t} \right)^2 \\ &+ EI \left(\frac{\partial u(x,t)}{\partial x} \right)^2 + GJ \left(\frac{\partial \gamma(t,x)}{\partial x} \right)^2 \end{aligned} \qquad (11)$$

in terms of flexion and torsion variables $u(t,x)$, $\gamma(t,x)$, and interaction part

$$\begin{aligned} \mathcal{K}_S =& \frac{1}{2} m X^2 + J_f \left(\frac{d\theta}{dt} + \frac{\partial^2 u(t,x)}{\partial x \partial t} \right)^2_{x=L} \\ &+ J_t \left(\frac{\partial \gamma(t,x)}{\partial x} \right)^2_{x=L} \end{aligned} \qquad (12)$$

$$X = (L + l_f)\frac{d\theta}{dt} + \frac{\partial u(t,x)}{\partial x} + l_f \frac{\partial^2 u(t,x)}{\partial x \partial t} + l_t \frac{\partial \gamma(t,x)}{\partial x}\bigg|_{x=L}$$

at links boundaries, out of which dynamical equations and boundary conditions are easily obtained[4]. (l_f, l_t) are coordinates of tip mass m with respect to link's end, and the various other coefficients characterize the beam as usual within Euler-Bernouilli approximation. One can verify that in link and actuator equations coupled by compliance effect, are both acting the applied input torque τ and the bending moment $M_a = EI(\partial^2 u(t,0)/\partial x^2)$, here the only term through which deformations are seen by system rigid part.

4-TASK ORIENTED CONTROL

In general, the system is assigned to perform an action, and a control is set to give the system the ability to meet the corresponding requirements. This is always expressed as satisfaction of Lyapounov theorem with adapted Lyapounov function, writen in terms of system trajectory parameters in state space. In other words, control is trajectory oriented, and all sensors are used in this view. In particular, vision sensor if any will provide information on link tip motion. As seen above, this is misleading as long as observed motion belongs to an indistiguishable class. Control has to be approached in task oriented sense, and, for reaching the goal, is governed by a choice of "good" informations depending of their utility defined above. Starting from partial Hamiltonian density associated to deformable part

$$\mathcal{H}_D = \dot{q}_j \frac{\partial \mathcal{L}_D}{\partial \dot{q}_j} + \dot{u}\frac{\partial \mathcal{L}_D}{\partial \dot{u}} + \dot{u}(S)\frac{\partial \mathcal{L}_D}{\partial \dot{u}(S)} + \dot{u}_\mu(S)\frac{\partial \mathcal{L}_D}{\partial \dot{u}_\mu(S)} - \mathcal{L}_D \qquad (13)$$

one will consider system Lyapounov function

$$\mathcal{V} = \int \mathcal{H}_D + \sum_j \left(K_{Pj}\frac{q_j^2}{2} + \Gamma_{Vj}\frac{\dot{q}_j^2}{2} \right) \qquad (14)$$

with positive parameter gains K_{Pj}, Γ_{Vj}. Its time derivative along system trajectories is

$$\begin{aligned} d\frac{\mathcal{V}}{dt} =& \sum_j \dot{q}_j \Big[U_j + F_j - \left(\frac{\partial L_R}{\partial q_j} - \frac{d}{dt}\frac{\partial L_R}{\partial \dot{q}_j} \right) \\ &+ K_{Pj} q_j + \Gamma_{Vj} \ddot{q}_j \Big] \end{aligned} \qquad (15)$$

Substituting for d^2q_j/dt^2 from explicited Lagrange equations(7) and eliminating all other second order time derivatives, one will get an "inertia" term F_{aj} which, on physical grounds, is equal to forces other than exterior forces F_j acting onto system of discrete variables q_j, and coming from the (back) effect of the field variables $u(x,t)$ onto discrete variables $q_j(t)$. As $\mathcal{V}$ is positive definite for large enough definite positive gains (K_{Pj}, Γ_{Vj}), its derivative can be made definite negative by taking control U_j so that the term between brackets is equal to $-(K_V\dot{q})_j$, where matrix K_V is definite positive. The resulting form of the control (supposing there is no exterior force)

$$U_j = -K_{Pj}q_j - K_V\dot{q}_j + K_D(q_j, \dot{q}_j, \cdots) + K_{Fj}F_{aj} \quad (16)$$

and generalizes usual PD-control to full nonlinear case. In fact, it fits more generally the expression of dynamical system control

$$U = U_{comp} + \overline{U}_{PDF} + \Delta U \quad (17)$$

when writing the tracking condition for desired trajectory $q_j(t) = q_{jd}(t)$ and splitting the various control components, with

$$U_{PDF} = \overline{U}_{PD} + K_F \begin{bmatrix} 1 \\ 0 \end{bmatrix} F_a \quad (18)$$

Moreover, from argument above, the control law in eqn (16) gives both asymptotic tracking of desired trajectory for discrete variables and asymptotic stability for field variables as well as their first order time derivatives.
From eqn(15), equating the sum between brackets in its right hand side to $-(K_V\dot{q})_j$ amounts to take a controller of PDA type[5]. However, it should be observed that the resulting invariant subset of $d\mathcal{V}/dt$ is the same as when $\Gamma_j = 0$. So the same convergence property of the solutions is expectable for any value of Γ_j. The reason of introducing the new kinetic term with $\Gamma_j \neq 0$ is in the role of the direct acceleration term, or of the new resulting term F_{aj} after substitution, which is mainly to change the relative values of inertia-damping-stiffness system parameters with respect to field modes, as already observed and used for classical force control.

But after substitution from Lagrange equations this term is an integral of a complicated function of field variables and their space derivatives over the domain $D(x)$. So there is no advantage to use it in this form which requires local knowledge of field variables inside the domain, unless Lagrangian structure is such that this integral transforms into explicit well identified and sometimes directly measurable surface quantities. A very simple case occur when the Lagrangian $\mathcal{L}_D$ is such that formally

$$\frac{\partial \mathcal{L}_D}{\partial q_j} - \frac{d}{dt}\frac{\partial \mathcal{L}_D}{\partial \dot{q}_j} = \frac{\partial \mathcal{L}_D}{\partial u} - \frac{d}{dt}\frac{\partial \mathcal{L}_D}{\partial \dot{u}} \quad (19)$$

Then from Lagrange eqns(7) there results

$$-\left(\frac{\partial L_R}{\partial q_j} - \frac{d}{dt}\frac{\partial L_R}{\partial \dot{q}_j}\right) = n_\mu . \left[\frac{\partial \mathcal{L}_T}{\partial u_\mu} - \partial_\nu \frac{\partial \mathcal{L}_T}{\partial u_{\mu\nu}} + \cdots\right]_{S_2}$$

The "inertia" force term F_{a2} is just equal to the boundary term in the first bracket of eqn(8) when $\Gamma_j = 0$, and is expressible in terms of this quantity, and of rigid variables q_j and their first time derivatives when $\Gamma_j > 0$. This global expression contains all needed information to control the local action of (infinite dimensional) deformation effects, usually approached by decomposing this source term onto all projection space and cutting at a finite mode number with spillover consequences[6,7].
Much more than local control, more global task oriented control will also be independent of (too) detailed information on link deformations. Typical task is to reach a preassigned target under specific circumstances. Returning to eqn(3), this amounts to minimize the total entropy production associated to any motion in the class of acceptable trajectories fixed by the local control defined in previous paragraph, so its expression depends in general of all trajectory parameters. To this end, the utility u will be taken as the gradient of a convenient positive definite quantity such as a Lyapounov function to define a steepest path and more importantly, to eliminate before data processing irrelevant task information, saving enormous amount of time and data space. So with (p) the set of all observed parameters one gets

$$u = \frac{\partial \mathcal{V}}{\partial p} \quad (20)$$

and in eqn(3) only will remain terms for which this expression is above a minimum threshold value corresponding to system sensitivity. So all collected information from sensors is filtered in terms of its utility for the prescribed task. This explicit result is independent of the dedicated or selfdeciding character of the system. With eqn(14) for instance, the only dependence of $\mathcal{V}$ on trajectory parameters is through bending moment M, so when taking the gradient with all sensor information, there only remains a term $\partial\mathcal{V}/\partial M$, and more detailed trajectory information does not appear. So adapted control splits finally into a local one expressed in terms of global (relative) invariants M, and a nonlocal one depending on utility of these quantities for reaching final target. Though trajectory oriented the first one directly links to the task oriented second one and respects the very nature of internal information provided by system structure. In this respect, system intelligence is easily measured by information flux from eqn(3) and by associated robustness ball of the applied control corresponding to a distance between demand and result.

5-CONCLUSION

Analysis of system structure shows that evaluation of its intelligence is only meaningful in task space. This requires the satisfaction of internal coherence conditions manifested by system ability to extract from its sensors the relevant information for these tasks. The problem is studied here by defining the useful information which precisely allow to pass from initial geometrical space to task space irrelevant of the way the system is designed and organized. Application is made for Lagrangian systems representing deformable bodies, for which equations analysis shows that even if at first sight system nature is drastically changing with increase of state space dimension to infinity, internal system organization also changes in such a way that its local control still remains fundamentally finite dimensional. Observation of new deformation modes is not only useless, but also damaging in that it leads to control form interfering with natural internal feedback regulating power exchange between displacement and deformation. Sensors providing too detailed information are not adapted as it has to be filtered for reconstitution of needed more global one. More efficient way is to use local control based on natural system invariants, directly linkable to more global task oriented control based on useful information (rather than filtered one) expressed in terms of utility factors constructed as the gradient of Lyapounov with respect to trajectory parameters. When they aggregate into trajectory invariants, only their derivatives finally appear, justifying again the choice of previous local control form. Moreover, the association of the two level form presented here respects natural system organization and minimizes information transfer between the two levels. System intelligence is directly measured by task adaptation expressed here as both circulating information flux and robustness ball corresponding to local controller for a given distance between demand and result.

References

[1] J. Khalfa, Ed. : *What is intelligence*, Cambridge University Press, Cambridge, Mass., 1994

[2] W.J. Book, T.E. Alberts, G.G. Hastings : "Design Strategies for High-speed Lightweight Robots", *Computers in Mechanical Engineering*, p.26,1986.

[3] M. Cotsaftis : *Comportement et Contrôle des Systèmes Complexes*, Paris, Diderot Multimédia, 1997.

[4] M. Cotsaftis : "Global Control of Flexural and Torsional Deformations of One-link Mechanical Systems", *Kybernetika*, Vol.33(1),1997,pp.75-86.

[5] P.T. Kotnik, S. Yurkovitch, U. Ozguner : "Acceleration Feedback Control for a Flexible Manipulator Arm", *Proc. 1988 IEEE Intern. Conf. on Robotics and Automation*, Philadelphia, Penn.,Vol.1,1988,p.322

[6] M.J. Balas : "Modal Control of Certain Flexible Dynamic Systems", *SIAM J. Control*, Vol.16,1978,p.450.

[7] Y. Sakawa : "Feedback Control of Second Order Evolution Equation with Unbounded Observation", *Int. J. of Control*, Vol.41(3),1985,p.713.

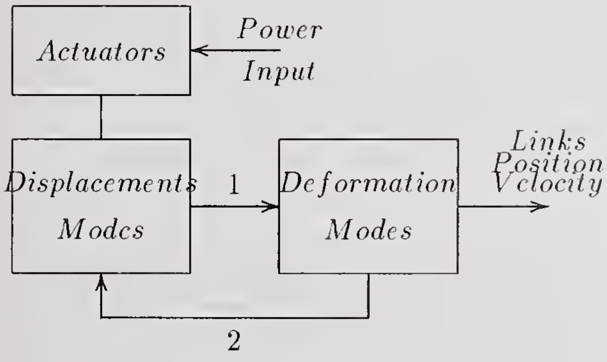


Fig.2 : Complex System Structure of Deformable Mechanical System

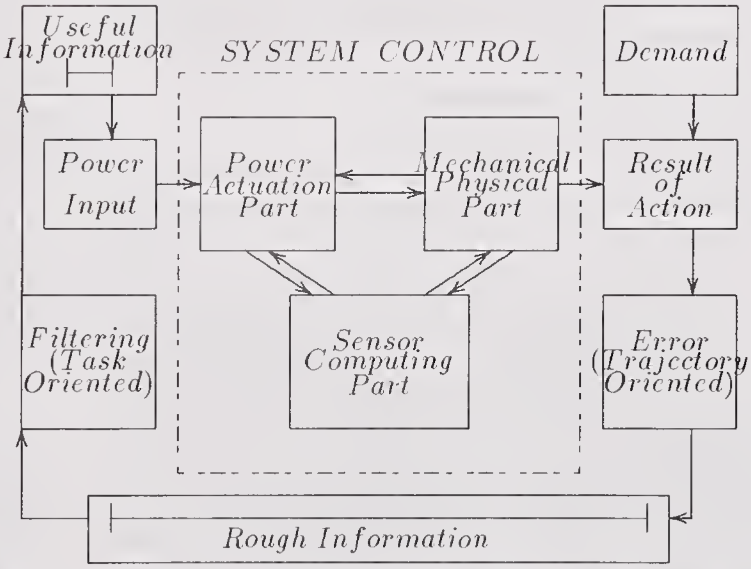


Fig.1 : System Structure with Main Component Parts and Information Filtering for Task Orientation Problem : Minimize distance(demand,result) =f(system parameters)

Minds, MIPS and Structural Feedback

Ricardo Sanz and Ignacio López

Universidad Politécnica de Madrid
{sanz,ilopez}@etsii.upm.es

Abstract

This paper tries to stress the need of having a clear understanding of the concept of intelligence before we can progress in the formulation of a measure for it. At the end it suggests a view of intelligence as *structural feedback in model-based control systems*.

Keywords: *Intelligence, performance, behavior, mental models, structural feedback.*

1 Introduction

This paper tries to suggest the practical impossibility of finding a *single* and *useful*[1] measure of general intelligence for all types of artificial systems performances unless we get some previous result in the form of a sound theory of intelligence.

As was stated at the workshop website, its goal is to discuss three challenges pertaining to intelligent system performance:

- how to measure performance;
- how to evaluate intelligence and
- how to put performance and intelligence into correspondence.

We will try to address the three points in order (see sections 4,5 and 6), but first we want to make a first comment. When talking about intelligence a problem appears, and it is that "intelligence" is a moving target. Some centuries ago "a person able to read" implied "a person very intelligent". Now we don't consider this ability as a symptom of intelligence in a person of our environment. But if we talk about an animal, forsone example a dog, "able to read" is still considered a good manifestation of intelligence.

So, what is that stuff that appears or disappears as you point at different entities? Can intelligence be in the eye of the beholder? We think that the term is used in two quite different ways: a) As a comparison between two entities that can be both explicit or one implicit (a normal dog) and b) As an absolute measure of some core capability.

While we can mostly agree with Alex Meystel conception of intelligence as a *core concept underlying minds*, perhaps all we are falling in the easy way of thinking mentioned by Bateson [3, page 82] of *using words that appear more concrete than they are*[2].

Before entering into main matter, let's start with a brief discussion about the adequacy of ascribing mental properties like *intelligence* to machines.

2 What is Intelligence?

It is common to address intelligence as a property inherent to something we call mind. The use of both terms, intelligence and mind, is not that clear. In fact, each one of us appears to have his own notion of intelligence speaking in terms of everyday life. Although deep thought and study about the topic can clarify partial notions of intelligence, there is still no global perspective.

We want the following question to emerge: *does intelligence really exist?* After what has been said and having in mind our constant references to the concept, it really seems ridiculous to question it. But we would like to point out the fact that *intelligence* could well be one word hiding what can be considered a too fuzzy concept [3]. By this we mean that the word does not have a fixed reference to something that can be pointed out, such as a dog or a table (it lacks a true referent). It is in some sense a concept similar to a notion of a mathematical space, i.e.: everything which matches certain restrictions is part of *intelligence*. The space of things that think.

The concept has lost in this way the apparent rigidity; the question, although, may be, in a more precise way: *what are the restrictions a feature has to match to form part of intelligence?* And at this point the answers diverge because the number of possibilities is close to infinity.[4] It would be an error to put the question like this. Perhaps it would be better to approach the topic in another way: *what is behind everything we seem to consider intelligent?* Searching this instead of a particular set of characteristics would eventually lead to a rule with which the judgement of the existance of intelligence would be possible.

In any case, once it is clear if something is intelligent or not, it would be tempting to determine *how intelligent*, that is, *how much intelligence it has*. This question is too particular to be

[1] From an engineering point of view, *i.e.* to build/analyze artificial systems.

[2] Bateson says about these words that they are too short and this shortness conveys an erroneous ascription of concreteness.

[3] A *linguistic variable* in its most pure sense: *i.e.* created by language.

[4] This is to be thought in a sense of *too broad for understanding*.

answered. The individual intelligent characteristics which constitute the *intelligent set of features* one self possesses are each specialised, and in this way not comparable.[5] In this way, given a set of intelligent characteristics, the only judgement that has any sense needs to be put in terms of targets and adequation to those targets: performance.

Returning to the rule which would enable discrimination between intelligent and not intelligent, it should not be focused on common aspects of features we usually consider intelligent, but on requirements which make them possible. For example parallel calculation, memory, etc. Having this in mind, the decision to consider something intelligent or not comes from the process of analysis of the underlying capability, i.e.: learning what can be expected from a being with such capability (eg. memory) when in a particular environment and with a more or less elaborate set of targets. Apparently we end again with a certain notion of performance.

The last point we would like to focus on comes from looking at the problem from a different angle. What if *intelligence* were a concept only suitable -clear enough- for human minds? That is, we call *something* intelligence, but it does not seem to have a bounded notion behind. So, supposing it is a collection of features we have grouped together, and not considering the fact that we could have done so in other ways, what makes us think that intelligence *is* something (a table, a bus)? In other words, what makes us think an alien would have a notion parallel to our *intelligence* as he would if he came to Earth and saw a table or a boat?

3 Human (species) chauvinism

Let's see what philosophers think about mental properties of machines. An example is what Crockett [5, p.193] says about the use of human–like phrases to refer to machine thinking:

> Our anthropomorphizing proclivity is to reify those abstractions and suppose that the computer program possesses something approximating the range of properties that we associate with similar abstractions in human minds.

Even more amazing is his continuation:

> This is harmless so long as we remember that such characterizations can lead to considerable philosophic misunderstanding.

What amazes me more in this text is that people like Crocket strongly believe that *we know* what are the "abstractions in human minds" but only *suppose* what the computer program possess. In our experience we know -most of the time- what are the abstractions -the representations- in mechanical minds but only suppose what are those abstractions in biological minds.

It is these days is when we are starting to get some direct insight into the inner working of human minds by means of PET (positron emission tomography) or fMR (functional magnetic ressonance [4]). As an example, fMR has confirmed what many had long suspected –that men and women think differently. Yale Medical School investigators did compare the brain operation of men and women while reading, discovering different activation patterns in their brains while performing the reading task.

Another example of the difficulties in matching human mental concepts with machine mental concepts can be found in [2]:

> Indeed, if mechanical devices can distinguish wavelengths of light without having sensations, then why do I experience any sensation at all?.

Most people tend to think that the human *sensation* is something more than the mere recognition of a input signal. Recognition at the simple level of signal capture, representation and triggering of activity. "Sensation" is nothing more than the triggering of activity due to an input signal. The immediate implementation in a computer is as an interrupt handler. The only difference is the high level of concurrence in biological computers that let them be truly concurrent in responses to sensations. There are also human sensations that are so strong that they disable further sensations. This is, exactly, the type of behavior found when a computer interrupt handler disables further interruptions.

Computers provide minds for physical systems, and it is time to clarify the true meaning of *mental concepts*.

4 Performance and MIPS in brains

A visible feature of biological intelligence is *performance* as Jim Albus pointed in his definition of intelligence. This is related to how we use the term for humans (remember the title of the book by Sternberg and Wagner, *Practical Intelligence: Nature and Origins of Competence in the Everyday World*).

In our search for metrics for intelligence, we are exactly in the same situation as computer consumers and manufacturers were some decades ago in relation with client-requested performance measures. As they both discovered, the old-basic measure of performance (MIPS: Million Instruction per Second) was useless to compare different architectures (*e.g.* CISC vs. RISC) or applications (*e.g.* data-bases vs. finite-element analysis). The only *useful* possibility they found was the evaluation of the performance in specific tasks, and hence this was the origin of benchmarking. Unfortunately benchmarks are not single measures, and attempts to build weighted benchmarks only changed the focus of the benchmark but not the final usability of them (they are always measures of niches of functionality).

Task-independent measures, like MIPS or *bits/second* or *entropy*, are too raw to be useful for most engineering purposes because they are so far from the desired performance specifica-

[5] It would be like comparing -adding, subtracting, etc.- apples and dogs: impossible.

tion that we lack a theory that can map one into another[6]. For example, suppose that we want a distillation column controller intelligent enough to minimize recirculation (a desired performance). Who can decide, based on a MIPS-like measure, if a fuzzy controller A can fulfill the task, or if model-based predictive controller B is better that A?.

This theory that maps a *MIPS-like measure* to *performance specified in useful terms* is what we are seeking in our research on intelligent systems, because it is –in fact– *The True Theory of Intelligence*. The theory will not only let us evaluate alternative designs, it will be a true explanatory discourse that will reduce intelligence to simpler, well grounded, terms.

To follow Bateson suggestion of marking concepts that are not concrete enough and require further thinking, we can use the term *i-stuff* to refer to the substance measured by True Intelligence Metrics. George Saridis probably will equate i-stuff to negentropy and Jim Albus to performance. We will make a suggestion at the end of the paper.

5 Intelligence and bodily capabilities

In relation with what can we measure, we agree with Chris Landauer in the fact that "Success is not by itself the right criterion" because we have to split success into two contributions: mind and body (and bodily intelligence is not what we are talking about). As an example consider two implementations of a future Mars rover whose main mission is going from point A to point B, one kilometer away, taking a sample of the ground each 50 meters:

Implementation H: 200 Ton. Caterpillar structure based on a combination of bulldozer, power shovel and truck. Control of sample taking based on mechanical coupling of power shovel to caterpillar (50 meters = sample). It lacks directional control because it is not necessary (it will advance straight *bulldozering* any obstacle.)

Implementation T: 50 Kilograms. 10 Watt solar power panel. Microrobotic arm.

Who will attain success? If both are successful, who is more intelligent? Is performance a manifestation of intelligence? The two first questions are rhetoric. The answer for the last one is "not always".

There are some attempts to extend fundamental physical theory to include information at the same level as mass and energy. In some sense we can analyze biological behavior as an exchange of mass (feeding in / excreting out), energy (chemical in / thermal & mechanical out) or information (sensing in / speech out). We can attach these interchanges to human subsystems, and information will become associated to the mental system. This division is, however, not very strict, because information is supported by means of mass or energy, and some energy inputs are managed as mass inputs (specially in animals).

6 Conclusions

Our analysis of the Mars rover story is that if the T implementation is successful everybody will agree that it is more intelligent than the H implementation. Even if both attain success. TO achieve this result the T implementation needs some mental content and some algorithms to exploit this mental content.

As we did say before we will propose a different interpretation of *i-stuff*: it is focused on *mental models*. Following this idea, an intelligent being is a being that has models of his world in his mind and achieves intelligent behavior using its models for action. Intelligence is, from this perspective, a two sided concept: model-based mental content (static view of intelligence) and model-based generation of behavior (dynamic view of intelligence).

Can the i-stuff be that collection models? Not so. Because all we know some knowledgeable people that are plain stupid.

What we consider the true core of intelligence is -plainly- feedback. When feedback for action is done trough good models of the world it achieves incredible performance levels. When feedback is used to tune parameter models it make systems adapt to changing circumstances in the world. When feedback is used to modify models of the world this is a pure learning process. When feedback is used to structurally modify the algorithms exploiting the models we arc talking of creativity[7]. *Structural feedback* is perhaps the highest manifestation of intelligence; when a system is able to create new control policies that will enhance its effectiveness.

Perhaps this proposal only muddles more the discussion because *model* is even shorter than *intelligence* and it seems even more concrete; but we think that it is relatively easier to devise metrics for model quality.

But even if we can measure quality of models and model evolution algorithms, we are still halfway to the metric of intelligent behavior, because we still lack a quality measure of the use of the model to generate the behavior (*i.e.* a metric of the architecture). Performance-based metrics, as suggested by Jim Albus definition of intelligence, will fit this niche but still they will be domain-dependent.

We strongly believe that, in the future, all these theories of intelligence will consolidate in a Great Unification Theory (and this *structural feedback* seems to us a good promising starting point), that will let engineers build artificial intelligences with the plasticity enough to adapt or tune to specific needs. Being this the case, in our opinion the core foundation of it will be raw information processing with capability to autoorganize in the form of models of the world and model exploitators generating behavior. The theory of intelligence can be viewed as a theory of action, a theory of representation or both.

[6] This is, in fact, the third point mentioned in the introduction.

[7] Adaptation, learning, evolution, creativity, are facets -i.e. perceptions from an external entity- of a system changing in response to interactions with the world.

References

[1] Christian Balkenius. *Natural Intelligence in Artificial Creatures*. PhD thesis, Lund University, Lund, Sweden, 1995.

[2] Stephen M. Barr. A mystery wrapped in an enigma. *First Things: A Monthly Journal of Religion and Public Life*, (77), November 1997. A comment on *The Conscious Mind: In Search of a Fundamental Theory*. By David J. Chalmers. Oxford University Press.

[3] Gregory Bateson. *Steps to an Ecology of Mind*. The University of Chicago Press, 1972.

[4] Neil R. Carlson. *Physiology of Behavior*. Allyn & Bacon, sixth edition, 1998.

[5] Larry J. Crockett. *The Turing Test and the Frame Problem. AI's Mistaken Understanding of Intelligence*. Ablex Publishing Corporation, Norwood, N.J., 1994.

[6] Kenneth M. Ford, Clark Glymour, and Patrick J. Hayes, editors. *Android Epistemology*. MIT Press, Cambridge, MA, 1995.

[7] Stanley P. Franklin. *Artificial Minds*. MIT Press, Cambridge, MA, 1995.

[8] Markus P.J. Fromherz, Vijay A. Saraswat, and Daniel G. Bobrow. Model-based computing: Developing flexible machine control software. *Artificial Intelligence*, 114(1-2):157–202, October 1999.

[9] Donald Gillies. *Artificial Intelligence and Scientific Method*. Oxford University Press, Oxford-New York, 1996.

[10] John Haugeland, editor. *Mind Design II*. MIT Press, Cambridge, MA, 1997.

[11] Tariq Samad. Complexity management: Multidisciplinary perspectives on automation and control. Technical Report CON-R98-001, Honeywell Technology Center, Minneapolis, MI, January 1998.

[12] Ricardo Sanz, Fernando Matía, and Santos Galán. Fridges, elephants and the meaning of autonomoys and intelligence. In *Proceedings of IEEE ISIC'2000*, Patras, Greece, 2000.

[13] Luc Steels and Rodney Brooks. *The Artificial Life Route to Artificial Intelligence*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1995.

[14] Kurt VanLehn, editor. *Architectures for Intelligence*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1991. The 22nd Carnegie Mellon Symposium on Cognition.

Fast Frugal and Accurate – the Mark of Intelligence: Towards Model-Based Design, Implementation, and Evaluation of Real-Time Systems

Bernard P. Zeigler and Hessam S. Sarjoughian
Arizona Center for Integrative Modeling & Simulation
Electrical & Computer Engineering Department
University of Arizona, Tucson, AZ 85721-0104, USA
Email: {zeigler|hessam}@ece.arizona.edu
URL: www.acims.arizona.edu

ABSTRACT

Engineered systems, whether called intelligent or not, principally must rely on models to achieve their goals even in the simplest situations. Therefore, a system's intelligence is a consequence of the collective intelligence embodied in its models. In this paper, we describe intelligence measurement grounded in the general concepts of discrete event, model-based system design methodology. We discuss the basic elements of the approach in view of their role in intelligence measurement. Computational resources in both processing and communication forms are constraints on intelligence, but they are not determinant The architecture which configures these resources plays a major role in the intelligence achieved. Further the architecture must support fast and frugal heuristics tuned to the environments in which the system is to operate. Real time processing architectures built on discrete event modeling and simulation principles are most suited to support "fast frugal and accurate" intelligence. Such architectures must be designed with a software engineering methodology that explicitly supports a system's control of its own computational resources and includes hooks for measuring its intelligence in terms of the speed, frugality and accuracy of its responses.

1 INTRODUCTION

Unless we are talking about the affluent life known to many of us in the recent past, the real world is a threatening environment where knowledge is limited, computational resources are bounded, and there is no time for sophisticated reasoning. Unfortunately, traditional models in cognitive science, economics, and animal behavior have used theoretical frameworks that endow rational agents with full information of their environments, unlimited powers of reasoning and endless time to make decisions. Tacitly accepting this paradigm – as seems the prevalent assumption – does not provide a promising basis for measuring intelligence, the theme of this conference.[1] Indeed, to measure intelligence requires first an understanding of the essence of intelligence as a problem solving mechanism dedicated to the life and death survival of organisms in the real world. Evidence and theory from disparate sources have been accumulating that offer alternatives to the traditional paradigm.

An important crystallization of the new thinking is the "fast frugal and accurate" (FFA) perspective on real word intelligence promoted by Todd and Gigerenzer [1]. FFA heuristics are simple rules demanding realistic mental resources that enable both living organisms and artificial systems to make smart choices quickly with a minimum of information. They are accurate because they exploit the way that information is structured in the particular environments in which they operate. Todd and Gigerenzer show how simple building blocks that control attention to informative cues, terminate search processing, and make final decisions can be put together to form classes of heuristics that have been shown in many studies to perform at least as well as more complex information-hungry algorithms. Moreover, such FFA heuristics are more robust than others when generalizing to new data since they require fewer parameters to identify.

It is important to note that FFAs are a different breed of heuristics. They are not optimization algorithms that have been modified to run under computational resource constraints, e.g., tree searches that are cut short when time or memory run out. Typical FFA schemes enable ignorance-based and one-reason decision making for choice, elimination models for categorization, and satisfying heuristics for sequential search. Leaving a full discussion of the differences to [1], the critical distinction is that FFA's are structured from the start to exploit certain restrictive assumptions, such as skewed frequency distributions, about their input data. They work well because these assumptions often happen to hold for data from the real world. Thus FFAs are not generic inference engines operating on specialized knowledge bases (the paradigm of expert systems) nor other generalized processing structures (e.g., [2]) operating under limited time and memory constraints. An organism's FFAs are essentially *models* of the real environment in which it has found its niche and to which it has (been) adapted.

New kinds of models for biological neurons provide possible mechanisms for implementing intelligence that is characterized by fast, frugal and accurate heuristics. Work by Gautrais and Thorpe [3] has yielded a strong argument for "one spike per neuron" processing in biological brains. "One-spike-per-neuron" refers to information transmission from

[1] NIST Workshop on Performance Metrics for Intelligent systems.

neuron to neuron by single pulses (spikes) rather than pulse trains or firing frequencies. A face recognition multi-layered neural architecture based on the one-spike, discrete event principles has been demonstrated to better conform to the known time response constraints of human processing and also to execute computationally much faster than a comparable conventional artificial neural net [4][2]. The distinguishing feature of the one-spike neural architecture is that it relies on a temporal, rather than a firing rate, code for propagating information through neural processing layers. This means that an interneuron fires as soon as it has accumulated sufficient input "evidence" and therefore the elapsed time to its first output spike codes the strength of this evidence. In contrast to conventional synchronously timed nets, in fast neural architectures single spike information pulses are able to traverse a multi-layered hierarchy asynchronously and as fast as the evidential support allows. Thorpe's research team has also shown that "act-as-soon-as-evidence-permits" behavior can be implemented by "order-of-arrival" neurons which have plausible real world implementations. Such processing is invariant with respect to input intensity because response latencies are uniformly affected by such changes. Moreover, coding which exploits firing order of neurons is much more efficient than a firing-rate code, which is based on neuron counts [3,4].

Countering the evidence that intelligence is essentially fast, frugal and accurate is Hans Moravec's prediction that by 2050 robot "brains" based on computers that execute 100 trillion instructions per second (IPS) will start rivaling human intelligence [5]. Underlying this argument is that there is an equivalence between numbers of neurons in biological brains and IPS in artificial computers. It takes so many billions of neurons to create an intelligent human and likewise so many trillions of IPS to implement an intelligent robot. In strong form this equivalence implies that pure brute force can produce intelligence and the structures, neural or artificial, underlying fast and frugal processing are of little significance.

2 MODEL-BASED INTELLIGENCE AND MEASUREMENT

In this section, we discuss intelligent systems from three perspectives: knowledge representation, execution, and measurement. Specifically, this paper makes the case that[3]

- computational resources in both processing and communication forms are constraints on intelligence, but they are not determinant
- the architecture which configures these resources plays a major role in the intelligence achieved
- the architecture must support fast and frugal heuristics tuned to the environments in which the system is to operate
- real time processing architectures built on discrete event modeling and simulation principles are most suited to support FFA intelligence
- such architectures must be designed with a software engineering methodology that explicitly supports a system's control of its own computational resources and includes hooks for measuring its intelligence based on FFA standards.

2.1 Computational resources in both processing and communication forms are constraints on intelligence, but they are not determinant

Morevac's claim that artificial intelligence will arise once the processing power is there to support it can be the starting point for a serious investigation to understand its merits. On the one hand, we need yardsticks of intelligence and on the other, yardsticks of computational resources (presuming that raw IPS is not very discerning). We might have a diagram as shown in Figure 1.

Let's assume for a moment that we have the framework in the form of a diagram as above, what can we do with it? We can ask

- For a given level of resources, how smart can a system be? This would prevent us from trying to build systems that are infeasible with the resources at hand.
- For a given intelligence level, how much resources are needed? This would help provide cost estimates for given intelligence requirements.
- How well does a system utilize its resources? Where does its intelligence stand relative to the best achievable in its resource league? Where does its level or resources stand relative to the best in its intelligence class?

[2] The face recognition layered net was executed by a discrete event simulator and took between 1 and 6 seconds to recognize a face on a Pentium PC vs. several minutes for a conventional net on a SGI Indigo. Recognition performance in both cases was very high. The authors employed a training procedure which, while effective, is not plausible as an in-situ learning mechanism.

[3] We are not claiming that these are the only elements responsible for intelligent behavior and by implication there are other means for intelligence measurement.

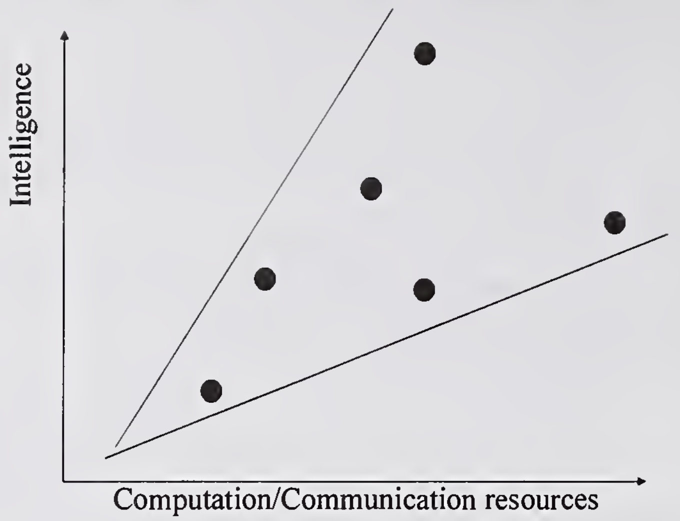


Figure 1: Intelligence measurement in terms of required resources

However, the yardsticks for resources and intelligence are not likely to be single dimensional linear orders but more likely to be multidimensional, partial orders. Even more to measure FFA intelligence which is environment-dependent, we may need to condition measurement with respect to problem classes asking which kinds of problems are performable on which kinds of architectures.

2.2 The architecture which configures these resources plays a major role in the intelligence achieved

This is a truism when applied to implementation of standard functionality – certain designs are better than others in implementing the same input/output behavior. However, in the absence of a well-defined characterization of intelligence in terms of input/output behavior, the focus has so far been on achieving intelligent behavior by whatever means possible, not paying much attention to the critical nature of the architectures that can support it. The results of Thorpe mentioned above, however, suggest that FFA intelligence is only achieved with "single-spike" neuron architectures and would be infeasible if the same neurons were employed in the manner assumed in conventional connectionist approaches.

2.3 The architecture must support fast and frugal heuristics tuned to the environments in which the system is to operate

Generalizing the idea that FFA heuristics embody models of the environment, the ability to work with models of the environment, one's self and others may be taken as key component of intelligence. Model-based design was formally introduced around 1980s as the basis to enable systems to reason about their own behavior in normal as well as abnormal situations. Over the years, many architectures have been proposed and implemented most of which typically suitable for narrow well-defined domains. However, a generic architecture based on simulation modeling concepts was proposed by [6]. Briefly stated, generic model-based design provides a generally applicable architecture in which simulation and other engines execute models that embody what the system employs about its environment – both external and internal

2.4 Real time processing architectures built on discrete event modeling and simulation principles are most suited to support FFA intelligence

Discrete event models can be distinguished along at least two dimensions from traditional dynamic system models – how they treat passage of time (stepped vs. event-driven) and how they treat coordination of component elements (synchronous vs. asynchronous). Recent event-based approaches enable more realistic representation of loosely coordinated semi-autonomous processes, while traditional models such as differential equations and cellular automata tend to impose strict global coordination on such components. Discrete event concepts are also the basis for advanced distributed simulation environments, such as the High Level Architecture (HLA) of the Department of Defense, that employ multiple computers exchanging data and synchronization signals through message passing [7]. Event-based simulation is inherently efficient since it concentrates processing attention on events – significant changes in states that are relatively rare in space and time – rather than continually processing every component at every time step.

The DEVS (Discrete Event Systems Specification) formalism [8] provides a way of expressing discrete event models and a basis for an open distributed simulation environment [9]. DEVS is universal for discrete event dynamic systems and is capable of representing a wide class of other dynamic systems. Universality for discrete event systems is defined as the ability to represent the behavior of any discrete event model where "represent" and "behavior" are appropriately defined. Concerning other dynamic system classes, DEVS can exactly simulate discrete time systems such as cellular automata and approximate, as closely as desired, differential equation systems. This theory is presented in [8, 10]. It also supports hierarchical modular construction and composition methodology [11]. This bottom-up methodology keeps incremental complexity bounded and permits stage-wise verification since each coupled model "build" can be independently tested.

An abstraction is a formalism that attempts to capture the essence of a complex phenomenon relative to a set of behaviors of interest to a modeler. A discrete event abstraction represents dynamic systems through two basic elements: discretely occurring *events* and the *time intervals* that separate them (Figure 2). It is the information carried in events and their temporal separations that DEVS employs to approximate arbitrary systems. In the quantized systems [8], events are boundary crossings and the details of the trajectories from one crossing to another are glossed over with only the time between crossings preserved.

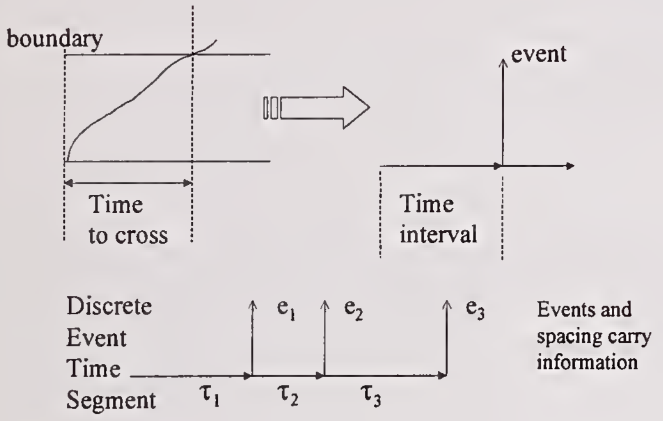


Figure 2: Discrete event representation of continuous trajectories

Recent results on discrete event neurons[4] show that, using a race analogy, a net of simple discrete event neurons can find the shortest path in a graph in the shortest time possible. Here is an instance where fast and frugal is provably optimal! In contrast, finding the longest path (or a long path) is much more difficult and requires much more sophisticated neurons. It seems uncanny – indeed, counterintuitively so – that minimizing performance measures such as distance, time, or cost requires simple apparatus and can be done with full accuracy and without backtracking. As with FFA heuristics, the mystery disolves when one recognizes that the discrete event neural nets exploit the underlying nature of reality in which pulses compete in parallel, and where fast competitors come first and lock out their slower countrparts from further progress. In the real world, fast response is paramount[5] and so minimizing time (or other meausers mapped into it) is critically important to survival. So brains may have been evolved to solve survival-critical problems with frugal means (simple neurons) that embody race analogies. Finally we note that discrete event neurons and one-spike-per-neuron architecutres are necessary to embody the race analogy – other models do-not work.

2.5 Such architectures must be designed with a software engineering methodology that explicitly supports a system's control of its own computational resources and includes hooks for measuring its intelligence based on FFA standards

Based on a wealth of basic research in a variety of disciplines, model-based design offers not only well-defined principles to design intelligent systems, but also can provide the means to assess a system from its inception to realization, operation, and eventual retirement. For example, we can assess a system's correctness, performance, maintenance, and cost, all of which are reflections of a system's degree of intelligence. We may also rank a system degree of intelligence in terms of, for example, intelligence of embodied models and how intelligently physical resources (computational and communication resources) are used.

Model-based design suggests several ways to rank intelligent systems based on their use of models:

- Distributed heterogeneous model-based architectures rank higher than monolithic ones.
- Systems that employ models that are at a resolution level compatible with the resources available to interpret them rank above those that don't.
  - □ Model sets that include self-representation rank above those that don't
  - □ Model sets that include representation of self and others rank above those that include only self-representation.
- Other rankings may be based on
  - □ Model abilities to handle both non-linguistic and linguistic queries
  - □ System ability to maintain coherence in the model base
  - □ System ability to inform meta-level models by questioning lower level models
  - □ Recursive depth of the "models-of" relation.

Due to increasing complexity and size/scale of systems (e.g., distributed agent-oreinted systems), it is becoming imperative to follow well-defined software development processes (e.g., waterfall, spiral, iterative, and/or incremental process [12]). A typical software development process is composed of conceptualization, analysis, design, implementation, and testing, and operation [13]. Indeed, the development of many contemporary distributed, heterogenouos systems must increasingly rely on such development processes [14]. Furthermore, with the emergence of archiecture-based paradigms, we can begin to devise suitable architectures for intelligent systems [15]. The archtiecture based apporach and software development processes go hand in hand offering many invaluable advantages such as incremental analysis, design, and testing. We believe, with the adoption of a synergistic development process (accounting for software, hardware, and bioware) combined with an appropriate architectural paradigm, we can incorporate, among other things, intelligence capabilities, metrics, and measurement methods in appropriate places.

[4] We are currently writing these results for publication.

[5] This is certainly a characteristic of e-commerce at internet speed.

3 ACKNOWLEDGEMENT

This research has been supported in part by NSF Next Generation Software (NGS) grant #EIA-9975050.

4 REFERENCES

[1] Gigerenzer, G., P.M. Todd, (1999). Simple Heuristics That Make Us Smart, Oxford University Press.

[2] Meystel, A.M., (2000). Simulation for Meaning Generation: Multiscale Coalitions of Autonomous Agents, in Discrete Event Modeling and Simulation Technologies: A Tapestry of Systems and AI-based Theories and Methodologies, Editors: H.S. Sarjoughian, F.E. Cellier, Springer Verlag.

[3] Gautrais, J. T. Simon, (1998). Rate coding versus temporal order coding: a theoretical approach, Biosystems (48)1-3, pp. 57-65

[4] Rufin, V.R., J. Gautrais, A. Delorme, T. Simon, (1998). Face processing using one spike per neuron, Biosystems (48)1-3 pp. 229-239

[5] Hans Moravec (1999). Rise of the Robots, Scientific American, August 1999, pp. 124-132

[6] Zeigler, B.P., (1990). Object-Oriented Simulation with Hierarchical, Modular Models: Intelligent Agents and Endomorphic Systems., San Diego, CA: Academic Press

[7] Fujimoto, R. (1998). "Time Management in the High-Level Architecture." Simulation 71(6): 388-400.

[8] Zeigler, B.P., H. Praehofer, and T.G. Kim, (2000). Theory of Modeling and Simulation. 2ed, New York, NY: Academic Press

[9] Zeigler, B.P., et al. (1998). The DEVS/HLA Distributed Simulation Environment and its Support for Predictive Filtering, ECE, The University of Arizona.

[10] Zeigler, B.P., et. al. (1997). "The DEVS Environment for High-performance Modeling and Simulation." IEEE CS&E 4(3)

[11] Zeigler, B.P. and H.S. Sarjoughian (1999). Support for Hierarchical Modular Component-based Model Construction in DEVS/HLA. Simulation Interoperability Workshop, Orlando, FL

[12] Pressman, R. (1997). Software Engineering: A Practitioner's Approach, McGraw Hill

[13] Booch G. (1996). Object Solutions: Managing the Object-Oriented Project. Menlo Park, CA, Addison-Wesley

[14] Orfali, R., D. Harkey. (1997). The Essential Client/Server Survival Guide, John Wiley & Sons

[15] Sarjoughian, H.S. and B.P. Zeigler (2001). "A Layered Modeling and Simulation Architecture for Agent-based System Development." IEEE Proceedings (to appear)

The Intelligence of an Entity

Robby Glen Garner
Steven Boyd Henderson

Preface

Mimetic Synthesis is a new terminology that more accurately describes a programming methodology used to mimic human behavior in a computer such as a PC. Previous work in this field has been incorrectly categorized under various aspects of Artificial Intelligence (AI).

On Intelligence

Testing and quantifying intelligence is difficult at best, even if it's human intelligence. To Quote Tariq Samad from "Notes on Measuring Intelligence in Constructed Systems", "The difficulty of compressing the multifaceted nature of intelligence into one scalar quotient has led to proposals to consider intelligence not as one unitary quantity but as a collection of properties that are mutually incommensurable." Furthermore, one of the many lessons from a century of work on human intelligence is that we still don't really know what intelligence is.

Mimetic Entities

The early mimetic systems developed by Robby Garner are hierarchical in structure. This allows the "Mimetic Entity" to synthesize the combined behavior of subsystems into a unified presentation. This structure certainly suggests that one way to measure the intelligence of such machines is to review the hierarchical concepts it uses and the processes that contribute to the goals of the whole system.

One of the first hierarchical mimetic synthesizers was called Albert. This program combined the behavior of several methods that shared the same goal of simulating human conversation. Each method represents a separate strategy used to form the response to a human stimulus phrase.

The first method is based on a simple model of behavior, where conversation is represented by strings of (stimulus → response) nodes. The goal of this particular method is to find a match for the user's input stimulus in a database, and form the reply with the corresponding "response" from the database. If the first method is not successful, the program follows down the hierarchy from most specific method, to least specific.

The second method looks in a table of Boolean rules and attempts to fit a rule to the user's input. If a rule is satisfied, its corresponding response is used. The goal of this method is to satisfy a Boolean expression based on the user's input phrase.

And so on, the third method attempts to find a generalization about the user's input phrase using a "framed" template to determine a match. The goal of this method is to find a generalization that applies to the user's input phrase.

Then finally, if none of the other methods has succeeded, a final method selects a "new topic" from a pool of unused topics. The goal of this method is merely to make a response. (To change the subject)

So, one can see that the overall goal of simulating conversation is attempted by using a variety of strategies, all contributing to the main goal. The hierarchical structure ensures that the best possible response may be used.

It must be obvious that the performance of the mimetic entity with regards to simulating a conversation depends entirely on the performance of all of these various methods or subsystems. Yet it depends first and foremost on the person talking to it.

The Loebner Show

But what can we say about Albert's intelligence? None of the methods used are intelligent, so their "unified" representation is not intelligent. Albert may be perceived as intelligent by a human being as is evidenced by the 1998 Loebner Prize Contest, but the program is not in fact intelligent. http://www.cs.flinders.edu.au/research/AI/LoebnerPrize/

Then if we can know what intelligence is not, does that tell us what intelligence is?

No, because none of the competitors in the Loebner contest have exhibited intelligence. At best they exhibit a behavior which seems familiar to the user (judge), and some of them have used very cleaver means to achieve this. But the ingenuity of the programmer does not make the program intelligent.

One also has to agree that an imitation is not the same as the thing it imitates. Furthermore, some may object to things that are artificial for no other reason except that they are artificial. Yet if a thing works, does it matter why it works or what it is made from? Some people would say that if a thing is not really "intelligent" then it is an impostor, and therefore "dangerous." But if a tool performs a job according to specification, why is that less intelligent than if a human being had performed the same job?

By doing a job, there is at least one goal implied, and that is the completion of the job. If a computer completes the same job as a human in a smaller amount of time, we would say the computer has better performance, not better intelligence?

Human Intelligence

In dealing with other people, we assess their intelligence on a casual basis by observing their behavior, the things they say, their solutions to problems, or other factors, many of which are purely subjective.

Measuring machine intelligence would be much easier if people could agree on how to measure human intelligence!

So I think there is always a disparity between "perceived intelligence" and "actual intelligence", especially in evaluation of human intelligence. Intelligence is not solely performance, but is it possible to measure intelligence without also measuring a performance?

Sometimes a performance involves a great deal of preparation and training. If a man repeats the same sequence of behavior, practices it over and over until it can be done repetitively without thinking, is that intelligence?

Summary

The key to true intelligence is the ability of an entity to enlist strategy to accomplish its mission, not preconceived knowledge, or rote behavior.

Military confrontation is a good example according to R. Neil Bishop. “Time and time again, superior firepower and resources have been overcome by an inferior force with an intuitive strategy, which gave them a monumental advantage.”

Also strategy is the key element needed to develop successful research techniques which, in pure science, may not even exist before the scientist begins. The strategy of obtaining and integrating knowledge is the key to reaching beyond what is presently known or understood.

The use of strategy applies not only to the highest level of abstraction, but is also evident in the “rank and file” subsystems that perform even the most basic tasks required by an entity as a whole. The strategy or algorithm employed by a programmer may be akin to “instinct” in some systems. Is instinctive behavior intelligent?

PART II
RESEARCH PAPERS

2. METRICS FOR INTELLIGENT SYSTEMS

Performance Metrics for Intelligent Systems

John M. Evans and Elena R. Messina
Intelligent Systems Division
National Institute of Standards and Technology
Gaithersburg, MD 20899-8230

ABSTRACT

Research into intelligent systems and intelligent control is burgeoning. However, there is no consensus on how to define or measure an intelligent system. This lack of rigor hinders the ability to measure progress in the field and to compare different systems' capabilities. We discuss some of the challenges and issues in defining performance metrics for intelligent systems and issue a call to action to participants in the Performance Metrics for Intelligent Systems Workshop to define practical metrics that will advance the state of the art and practice.

KEYWORDS: *performance metrics, intelligent systems, intelligent control*

1. Introduction

Intelligent systems are increasingly being identified as solutions to many advanced applications in manufacturing, defense, and other domains. Industry workshops [4] and roadmaps [3] specifically call for intelligent control or intelligent systems to address needs such as

- Adaptive, reconfigurable manufacturing equipment and processes
- Self-optimizing, science-based control of manufacturing unit processes
- "First part correct," that is, the ability to design and manufacture a product correctly, the first time and every time
- Self-diagnosing and self-maintaining systems
- Tool wear and breakage monitoring

Government agencies are basing major programs on intelligent capabilities, for example,

- The Army Experimental Unmanned Ground Vehicle Systems (Demo III)
- Defense Advanced Research Projects Agency (DARPA)/Army Future Combat Systems
- DARPA Mobile Autonomous Robot Software
- DARPA Software for Distributed Robotics
- DARPA Tactical Mobile Robots
- National Aeronautics and Space Administration (NASA) spacecraft and rovers
- Department of Energy (DOE) waste remediation robot systems
- Department of Transportation (DOT) Intelligent Vehicle Initiative

In addition to the examples above, there are myriad other efforts in academia, industry, and government labs of work referred to as "intelligent systems." Despite the common use of "intelligent system" and "intelligent control," there is no uniform definition for either term. Generally, they are characterized by having one or more of the following traits [1]:

- Adaptive
- Capable of learning
- "Does the right thing" or "acts appropriately"
- Non-linear
- Autonomous symbol interpretation
- Goal-oriented
- Knowledge-based

These terms are ambiguous and qualitative. The Intelligent Systems Division of the National Institute of Standards and Technology has launched an initiative to better define what an intelligent system is and how to measure its performance. The mission of the Intelligent Systems Division, one of five divisions in the Manufacturing Engineering Laboratory, is "to develop the measurements and standards infrastructure needed for the application of intelligent systems by manufacturing industries and government agencies."

We are working with various industry groups and government agencies to tackle the issue of intelligent system performance. The Performance Metrics for Intelligent Systems Workshop is a foundational step, which brings together a multi-disciplinary community to help define the highest priority areas to concentrate on, having the highest payoff.

2. The Challenge of Defining and Measuring Machine Intelligence

Researchers have been pursuing forms of machine intelligence for several decades. There have been many areas of focus, such as natural language understanding, expert systems to aid diagnoses, and decision-making tools for financial systems. Closer to our domain of interest, much effort has been focused on defining intelligent control as a discipline, but even so, there are no quantitative measures.

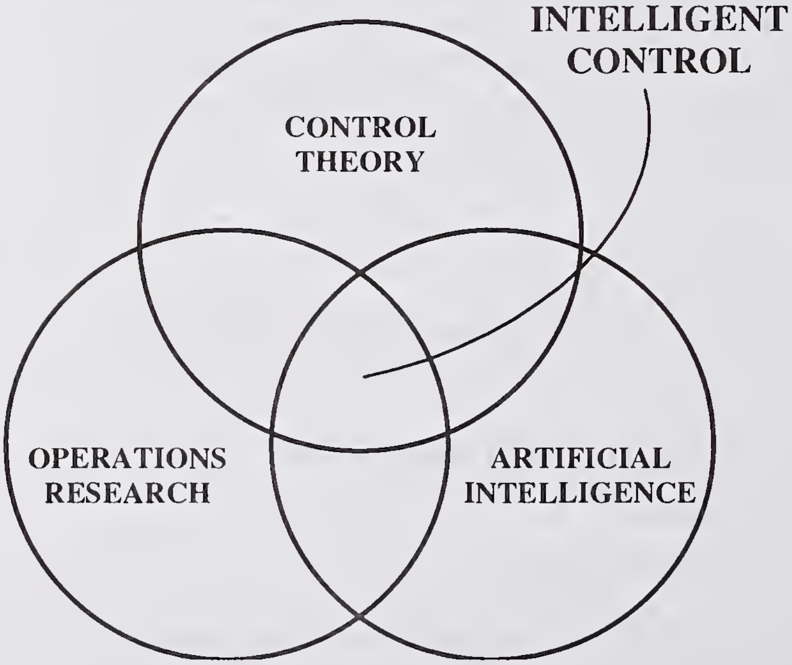


Figure 1: Intelligent Control as of 1985

Beginning with the efforts of Fu [1] and Saridis [3] in the seventies, there have been numerous conferences and workshops aimed at the topic of intelligent control. Nevertheless, the field remains fragmented due to its multidisciplinary nature. As noted in the first Symposium on Intelligent Control in 1985, intelligent control was proclaimed a theoretical domain, in which control theory, AI, and operations research intersected (Fig. 1 from [6]).

The definition of an intelligent system may be considered broader than that of intelligent control. As a "system," there may be more constituent parts, such as perception, world modeling, or value judgement. Yet more disciplines are brought into the picture. Examples of these include data representation, image processing, and decision theory.

Given the multi-disciplinary nature of the systems we are concerned with, it is clear that *defining the scope and performance of these systems is a challenge.* Terminology is one of the first hurdles that must be overcome. Different disciplines ascribe different definitions to the same words. For example, "complexity" may refer to non-linear systems in one field and to computational resources needed in another.

It is very difficult, if not impossible to currently evaluate research into intelligent systems. Since there are no quantitative metrics, intercomparisons of results are not generally possible. Sponsors are not able to adequately judge whether research results meet their requirements. Potential users have no impartial evaluation reports, *a la* "Consumer Reports," of intelligent systems, techniques, and tools. In general, the lack of metrics slows progress. There is a proliferation of data specific algorithms and task-specific solutions.

One of the biggest costs paid is the duplication of effort. New programs may be unable to have a firm definition of past accomplishments, hence they may fund work that repeats previous research. Research teams cannot leverage prior existing work from other institutions and tend to have to reinvent the wheel by building all of their system's components from scratch. They are burdened with having to spend effort in building components that are not part of their research focus, instead of being able to leverage existing "best of class" solutions and focussing on their interests.

Another negative impact, from the sponsor's viewpoint, is the lack of predictive ability in assessing new applications. Without objective performance evaluation metrics and an understanding of capabilities and limitations, it is difficult or impossible to assess claims of competing approaches in formulating new projects and programs. This leads to inefficiencies and failures that could be avoided if we had the measurement tools that we need.

3. Issues in Measuring Performance

Numerous questions must be answered when considering how to define the performance of these intelligent systems. We will present a few questions. Many more will arise as we delve into the matter more closely.

- Should we measure only the external behavior of a system? Is that the only aspect that can feasibly be measured? Or, is there value in decomposing a system into components and measuring their individual capabilities? Examples would be measuring the path planning algorithms in isolation from the perception and other control subsystems.

- How generic does the measure of a system's intelligence have to be? Should we strive for general intelligence metrics that are domain-independent or are we better off focussing on application and domain-specific metrics? Are domain-independent metrics even meaningful?

- How do we factor in "body intelligence," the mechanical capabilities of a system as opposed to the control capabilities, when assessing the performance of a system? If we have a mobile robot, some of its abilities to achieve its stated goal (e.g., traverse a rubble pile to find survivors) can be attributed to its mechanical properties rather than its software intelligence.

- Are testbeds a viable measure of performance, or do they invite "gaming," that is, encourage solutions that are tailored to performing well in the testbed? If we don't have testbeds, how can we achieve reproducible measures of performance?

4. Initial observations

One of the complicating factors in discussing intelligent systems is the use of the word "intelligence." It is freighted with significance and analogies to human or biological intelligence naturally arise. The quest for standard, uniform measures of intelligence in biological systems remains a subject of controversy. Therefore, we would advocate avoiding the temptation to spend too much time striving for performance measures that are based on human or higher level biological systems.

Observing that we are dealing with multi-disciplinary technologies and multiple application domains, we should expect that no single, unique measure of performance is feasible. Therefore, no single overarching and generic intelligence test will suffice. We need to strive for the right granularity of metrics.

We must be prepared to attack the problem on multiple fronts. It probably won't suffice to have just a theoretical investigation or an experimental one. Research must proceed on the theory as well as on gathering experimental data.

One of the key attributes of intelligent systems is its multi-disciplinarity. This poses a challenge, but also an opportunity. We can come together from a variety of disciplines and form a new community in which we share our expertise. We must have dialog and information exchange amongst ourselves in order to synthesize the best results from the different fields that contribute towards intelligent systems research.

That is the purpose of this workshop and the reason for the diversity of the presentations that you will hear.

5. Call to action

The challenge is thus to define performance measures for new and evolving intelligent systems technologies that can greatly improve industrial productivity and advance government mission objectives. We must work together to build a technical foundation for measuring performance. This includes agreeing on the domains to investigate and a common set of terminology. We must develop theoretical foundations, methodologies, and supporting infrastructure for achieving our goals. Ultimately, measures must be developed that are practical, unambiguous, easy to use and widely deployable. We must simultaneously focus on attainable goals and strategies for both near-term and long-term measures of performance, as our understanding of them and the capabilities of the systems themselves evolve. Researchers, industry, and government will benefit from practical solutions they can readily apply, not from philosophical ones.

6. References

[1] P. J. Antsaklis, "Defining Intelligent Control", Report of the Task Force on Intelligent Control, P.J Antsaklis, Chair, IEEE Control Systems Magazine, pp. 4-5 & 58-66, June 1994.

[2] K.-S. Fu, "Learning Control Systems and Intelligent Control Systems: An Intersection of Artificial Intelligence and Automatic Control," IEEE Trans. on Automatic Control, Vol. AC-16, 1971.

[3] Integrated Manufacturing Technology Roadmapping Initiative, “”Manufacturing Processes and Equipment Draft Roadmap,” Oak Ridge, TN, December, 1998.

[4] Neal, R. and Messina, E., ed., Proceedings of the First Part Correct Workshop, Gaithersburg, MD, April 2000.

[5] G. Saridis, Self-organizing Control of Stochastic Systems, M. Dekker, NY, 1977.

[6] G. N. Saridis, "Foundations of the Theory of Intelligent Controls," Proc. of the IEEE Workshop on Intelligent Control, Eds. G. Saridis, A. Meystel, Troy, NY 1985, pp. 23-28.

July 18, 2000

The Search for Metrics of Intelligence -- A Critical View

Lotfi A. Zadeh*

Few issues in AI generate as much heated debate as those which in one way or another relate to the questions: "What is intelligence?"; "Can machine think?"; and "How can intelligence be measured?" One cannot but be greatly impressed by the incisive comments made by members of the Intelligence Advisory Board. And yet, most of the basic issues relating to intelligence remain unresolved -- as they were half a century ago -- when I moderated, at Columbia University, what I believe to have been the first debate on "Can machines think?" The debate involved Claude Shannon, E.C. Berkeley, the author of Giant Brains, and Professor Francis J. Murray -- a prominent mathematician who as a consultant to IBM was active in the conception and design of computer systems.

At that time -- the dawn of the computer age -- there was a great deal of interest in the ability or inability of computers to think as humans do. To a much greater degree than is the case now, there were exaggerated expectations. In an article of mine entitled "Thinking machines -- a new field in electrical engineering," which appeared in the January, 1950, issue of the Columbia Engineering Quarterly (Zadeh 1950), I surveyed some of the articles which were published in the popular press at that time. The headline of one of the articles read "Electric brain capable of translating foreign languages is being built." The problem of machine translation seemed to be

* Professor in the Graduate School and Director, Berkeley Initiative in Soft Computing (BISC), Computer Science Division and the Electronics Research Laboratory, Department of EECS, University of California, Berkeley, CA 94720-1776; Telephone: 510-642-4959; Fax: 510-642-1712; E-Mail: zadeh@cs.berkeley.edu; Webpage: http://www.cs.berkeley.edu/People/Faculty/Homepages/zadeh.html.
Research supported in part by ONR Grant FDN0014991035, NASA Grant NAC2-1177, ONR Grant N00014-96-1-0556, ARO Grant DAAH 04-961-0341 and the BISC Program of UC Berkeley.

close to solution. Today, we know better. In 1997, Martin Kay, one of the leading contributors to machine translation had this to say: "Machine translation gave the initial inspiration to computational linguists and continues to motivate much of their work. That is surely fair enough since the problem is clearly computational and obviously linguistic. But forty years of money and effort has brought us hardly any closer to the answer. The world continues to pour money down the same rathole with little discernible progress, with or without the linguists. The German government is giving it a new twist: "Notice how we never seem to get anywhere on machine translation?"

The debates which raged in the past were largely of academic interest because there were few, if any, systems that could be assessed as having a high level of intelligence. At this juncture, this is no longer the case. Today, we can point with pride to Deep Blue, which beat Gary Kasparov. More importantly, we have a wide variety of systems which can perform highly non-trivial tasks involving recognition, decision and control. We are, in fact, witnessing the beginning of what may be described without exaggeration as the Intelligent Systems Revolution.

When AI was christened in 1956, it became the standard bearer of efforts to devise and build machines that could exhibit human-like intelligence in performing various tasks. For some time thereafter, the AI scene was one of unbridled enthusiasm and, as we now realize, unrealistic expectations. In judging that period, however, what should be remembered is that -- as Jules Verne astutely observed at the turn of the century -- scientific progress is driven by exaggerated expectations.

It took forty years for a computer to challenge and beat a chess champion. Why did it take so long to achieve some of AI's objectives? In the first place, the basic difficulty of approximating to what humans can do so easily without any measurements and any computations, e.g., understand speech, read handwriting, summarize a story and park a car, was greatly underestimated. More important, however, is the fact that the needed technologies and methodologies were not in place. In particular, we did not have the highly capable sensors and powerful computers which we have today, and we did not employ such recently developed methodologies as neurocomputing, evolutionary computing, probabilistic computing, machine learning and fuzzy logic.

In the past, what were called intelligent systems were for the most part symbol-manipulation oriented, e.g., machine translation systems, text understanding systems and game playing systems, among others. Today, what we see is the rapidly growing visibility of systems which are sensor-based and have embedded intelligence, e.g., smart washing machines, smart air conditioners, smart rice cookers and smart automobile transmissions. The counterpart of the concept of IQ in such systems is what might be called Machine IQ, or simply MIQ (Zadeh 1994). However, what is important to recognize is that MIQ -- as a metric of machine intelligence -- is product-specific and does not involve the same dimensions as human IQ. Furthermore, MIQ is relative. Thus, the MIQ of, say, a camera made in 1990 would be a measure of its intelligence relative to cameras made during the same period, and would be much lower than the MIQ of cameras made today.

Viewed in this perspective, the focus of activity in applications of machine intelligence is shifting from writing computer programs that can prove difficult theorems, understand text, provide expert advice and beat a chess champion, to more mundane tasks devolving on the conception, design and construction of products and systems that have a high MIQ, making them reliable, capable, affordable and user-friendly. Among recent examples of systems of this kind are programs which can detect the presence of known or new viruses in computer programs; checkout scanners which can identify fruit and vegetables through the use of scent sensors; car navigation systems which can guide a driver to a desired destination; password authentication systems employing biometric typing information; ATM eyeprint machines for identity verification; and molecular breath analyzers which are capable of diagnosing lung cancer, stomach ulcers and other diseases.

If MIQ is accepted as a metric of machine intelligence, then a particular machine may be said to be highly intelligent if has a high MIQ. But this beg the question of how the MIQ of a class of machines could be measured. Comments made by members of the Intelligent Advisory Board provide some guidelines. But a thesis that I should like to put on the table is that the existing conceptual framework of AI -- which is based on first-order two-valued logic -- is incapable of providing a suitable foundation for constructing realistic metrics of IQ and MIQ.

The problem with predicate-logic-based AI is that it embraces the principle of the excluded middle, which asserts that every proposition is either true or false, with no shades of gray allowed. But in the real world, as perceived by humans, it is partiality rather than categoricity that is the norm. Thus, we generally deal with partial knowledge, partial order, partial truth, partial certainty, partial causality and partial understanding. The essentiality of the role of partiality in human cognition has been slow in gaining acceptance in AI. Without employing the notion of partiality, realistic metrics of IQ and MIQ cannot be constructed.

Another concept that plays a basic role in human cognition is that of granularity, and, more particularly, that of f-granularity. In essence, f-granularity is a concomitant of the bounded ability of sensory organs and, ultimately, the brain, to resolve detail and store information. What this means is that (a) the boundaries of perceived classes are not sharply defined; and (b) values of perceived attributes are granulated, with a granule being a clump of values drawn together by indistinguishability, similarity, proximity or functionality. For example, the granules of Age might be: very young, young, middle-aged, old and very old. Similarly, the granules of face may be: nose, cheeks, chin, forehead, etc. F-granularity underlies the concept of a linguistic variable in fuzzy logic.

The concepts of partiality and f-granularity play key roles in what may be called Precisiated Natural Language (PNL). What I should like to suggest is that PNL could play a central role in formulation of metrics of intelligence. How these could be done is a complex task that will require a major effort to yield concrete results. In what follows, I will confine myself to sketching the basics of PNL and pointing to its use as a concept definition language.

Natural languages are expressive but imprecise. Mathematical languages are inexpressive but precise. Basically, PNL draws on a natural language (NL) and a mathematical language (ML) to provide a language which is precise and yet far more expressive than conventional meaning-representation and definition languages based on predicate logic.

In essence, PNL is a subset of NL which consists of propositions which are precisiable through translation into a precisiation language GCL (Generalized Constraint Language). An example of a precisiable proposition is: It is very unlikely that there will be a significant increase in the price of oil in the near future. The point of departure in PNL is the assumption that the meaning of a precisiable proposition, p, is expressible as a generalized constraint on a variable. Usually, the constrained variable and the constraining relation are implicit rather than explicit in p.

A concept which has a position of centrality in GCL is that of a generalized constraint expressed as X isr R, where X is the constrained variable, R is the constraining relation, and isr (pronounced as ezar) is a variable copula in which r is a discrete-valued indexing variable whose value defines the way in which R constrains X. Among the principal types of constraints are the following: possibilistic constraint, r=blank, with R playing the role of the possibility distribution of X; veristic constraint, r=v, in which case R is the verity (truth) distribution of X; probabilistic constraint, r=p, in which case X is a random variable and R is its probability distribution; r=rs, in which case X is a fuzzy-set-valued random variable (fuzzy random set) and R is its fuzzy-set-valued probability distribution; and fuzzy-graph constraint, r=fg, in which case X is a fuzzy-set-valued variable and R is its fuzzy-set-valued possibility distribution.

With these constraints serving as basic building blocks, which are analogous to terminal symbols in a formal language, more complex (composite) constraints may be constructed through the use of a grammar. Simple examples of composite constraints are: X isr R and X iss S; and, if X isr R then Y iss S, or, equivalently, Y iss S if X isr R. The collection of composite constraints forms the Generalized Constraint Language (GCL). The semantics of GCL is defined by the rules that govern combination and propagation of generalized constraints. These rules coincide with the rules of inference in fuzzy logic (FL).

The capability of PNL to serve as a powerful definition language depends in large measure on the fact that, by construction, GCL is maximally expressive. The conclusion that emerges from this fact is that metrics of intelligence, if they can be defined, will necessarily have to be defined in terms of PNL and have an algorithmic structure (Zadeh 1976). What this implies is that realistic metrization of intelligence is not possible within the conceptual structure of existing methods of definition and measurement. We cannot expect a concept as complex as that of intelligence to be definable in traditional terms.

References

1. L.A. Zadeh, "Thinking machines - a new field in electrical engineering," *Columbia Engineering Quarterly 3*, 12-13, 30, 31, 1950.

2. L.A. Zadeh, "A fuzzy-algorithmic approach to the definition of complex or imprecise concepts," *Int. Jour. Man-Machine Studies 8*, 249-291, 1976.

3. L.A. Zadeh, "Fuzzy logic, neural networks and soft computing," *Communications of the ACM 37(3)*, 77-84, 1994.

Measure of System Intelligence: An Engineering Perspective

Sukhan Lee[†], Won-Chul Bang[‡] and Z. Zenn Bien[‡]
[†] System and Control Sector, SAIT, Korea
[‡] Div. of EECS, Dept. of EE, KAIST, Korea

ABSTRACT

System intelligence can be measured experimentally either through benchmark tests, or theoretically through the formal analysis of system software architecture and hardware configurations. The latter approach is pursued here, since it serves directly as the criteria for designing and engineering intelligent systems in a directed manner, rather than by trial and error. To this end, a structure of problem solving and learning of machine is proposed. Once a machine is represented with the structure, the intelligence can be measured via transforming it into an equivalent linguistic structure. A simple example is also provided.

KEYWORDS: *measure of system intelligence, measure by linguistic equivalence, machine description language*

1. INTRODUCTION

The intelligence of systems is emergent when the systems are able to accomplish loosely defined but complex tasks in an unstructured and uncertain environment. The intelligence can be manifested by the capability of systems to autonomously synthesize goal-oriented behaviors in adaption errors, faults, and unexpected events through the real-time connection of sensing and action. However, we still do not have a satisfactory quantitative way to characterize the "intelligence" of systems. There are many kinds of intelligent systems in various fields. The adjective 'intelligent' is quite widely used to describe their systems developed by many system engineers and companies. One developer may say that his/her system is more intelligent than the others, but it can happen that another claims the same thing. In this case, who can say one is more intelligent than the others? One must have a kind of measure of intelligence for systems or machines in order to answer this question. In this sense, it is worthwhile to provide a measure on how intelligent a machine is.

Many intelligent system techniques have been developed and studied so far, but only a few studies have been done on '*how to measure intelligence of systems.*' J. S. Albus introduced the theory of intelligence in an engineering viewpoint [1]. G. Zames initiated an effort for defining such an index as approximate a measure of the "task" and "satisfactory" performances an "intelligent controller" could achieve versus those that a classical controller could achieve [2]. The challenge involves characterization of performance in unknown environments, learning, controller and task complexity, and associated tradeoffs. E. C. Chalfant and S. Lee suggested an engineering perspective [3]. They thought that one can represent all tasks of a machine in the form of graphs and find an equivalent language for the graphs. Since a language consists of grammar and vocabulary, the descriptive power of a machine can be represented by the grammar and the vocabulary. Bien, et al. [4][5] proposed a couple of methods to measure how much a machine is intelligent; they considered the questions from the ontological (functional) and phenomenological (behavioral) definitions on intelligent machine.

Establishing the measure of system intelligence should not only be able to turn the intelligent system into a formal academic discipline but also provide a means of designing better and more powerful intelligent systems in practice. The measure of intelligence of a system or, more precisely, a constructed system with autonomy should take into consideration various aspects of intelligence ranging from perception, understanding, and problem solving to generalization and learning from experience. A. Meystel proposed a vector of system intelligence as a collection of features representing intelligent functions of a system. The list of such features can be very comprehensive indeed. However, formulating the measure of system intelligence based on such a vector may not necessarily represent the essence of system intelligence. The functional features describing the aspect of intelligent behaviors may obscure the existing internal engine by which intelligent behaviors are generated.

To begin with, the following questions are raised for answer prior to the definition of the metric of system intelligence:

(a) Should the intelligence measure be goal-dependent or goal-independent?

(b) Should the intelligence measure be time-varying or time-invariant?

(c) Should the intelligence measure be resource-dependent or resource-independent?

For (a), it raises a question whether there exists a universal measure of system intelligence such that the intelligence of systems can be compared independently of the given goals. A goal-independent measure may be more difficult to define, if not impossible, and more controversial. A goal-dependent measure, however abstract the goal may be, can allow clear comparison among the systems of different architecture but with the same goal. For instance, for the latter case, intelligence can be represented as how efficiently, and how optimally a system reaches the given goal by itself, i.e., the power of automatically solving problems defined as the discrepancy between the goal and the current state.

For (b), it represents whether the intelligence measure of a system should solely be based on problem-solving capability at time *t* or it should contain the potential increase of problem-solving capability in the future based on learning. Both are necessary. But, it is better to define the two separately before integrating them together in one measure.

For (c), it raises an issue whether the resources required for building systems and system operation should play a role for defining the measure of intelligence. As mentioned above, the efficiency in problem solving should be included in the measure: for instance, the time and energy required to reach a solution should be taken into consideration together with the optimality of the solution. But, it is not clear whether we should or should not include the cost of building a system.

Section 2 provides definitions of engineering metric of system intelligence based on the above three questions. In Section 3, machine intelligence structure is proposed, and an equivalent linguistic structure follows in Section 4. Section 5 shows an example with a robotic arm. Finally, Section 6 concludes the paper.

2. DEFINITION OF ENGINEERING METRIC OF SYSTEM INTELLIGENCE

System intelligence can be measured under considering various points of views described in the previous section. An approach in engineering perspective is pursued here with *goal-oriented*, *time-dependent*, and *resource-dependent* definition of engineering metric of system intelligence. We define machine intelligence quotient (MIQ) in the following way.

The measure of system intelligence as problem-solving capability at time *t* for the given goal set *g*, denoted by *MIQ*(*g*, *t*), is defined by the capability of solving problems toward the given goal set where the capability can be measured by the scope of constraints (environmental variations), together with the time and resources required, under which the system succeeds in reaching the given goals.

The measure of self-improvement of system intelligence as learning capability with respect to time *t*, denoted by *dMIQ*(*g*, *t*), can be defined by the rate of increasing *MIQ*(*g*, *t*) with respect to time based on learning from experience. Capability of learning in the time duration of (t_1, t_2) is represented by the integration of *dMIQ*(*g*, *t*) between t_1 and t_2.

Now, the total measure of system intelligence, *tMIQ*, is defined by

$$tMIQ = \max_{t}[MIQ(g,t_0) + \int_{t_0}^{t} dMIQ(g,t)dt]. \qquad (1)$$

Let *tmax* be the time when the maximum of *tMIQ* is obtained. The learning rate is then defined by $\max_{t} \int_{t_0}^{t} dMIQ(g,t)dt \Big/ tmax$.

Note that the universal measure of system intelligence, *uMIQ*, may be defined in terms of integration of MIQ with respect to goal, i.e.,

$$uMIQ = \int_{g\in G} \max_{t}[MIQ(g,t_0)+ \int_{t_0}^{t} dMIQ(g,t)dt]dg \qquad (2)$$

where *G* is the set of all goals.

As mentioned above, resources required for the machine is combined into the machine intelligence, MIQ to resource ratio, *rMIQ*, can be represented by

$$rMIQ = tMIQ/resources. \qquad (3)$$

3. MACHINE INTELLIGENCE

As described in the previous section, machine intelligence can be measured once *MIQ*(*g*, *t*) and *dMIQ*(*g*, *t*) are defined. We now formulate the way of defining two quantities, *MIQ* (problem-solving capability) and *dMIQ* (rate of increasing *MIQ* based on learning capability).

The first step of problem solving is to understand the situation and define what are the problems to solve. This requires identifying the gap between the goal and current states as well as recognizing the constraints and opportunities imposed by the environment. Then follows the planning or decision-making to reduce the gap under constraints. The first step requires perception and understanding, whereas the second step requires action and planning. Perception and action can be represented as logical sensor and actuator systems, respectively, in a form of hierarchical graphs of declarative knowledge components. Understanding can be represented as the connection of what have been perceived to system internal knowledge. Planning can be represented as the projection of what have been understood to the logical actuator system. The mechanism of these connections can be rule-based. The overall structure of problem solving mechanism is represented in Figure 1 with solid-line connections.

Regarding the learning capability, a higher level of consciousness that monitors these activities of understanding and planning may exist in the form of thinking (a self-driven function that monitors understanding and planning in the form of questioning, virtual manipulation). In case that the machine cannot understand an obtained data from logical sensors by perception, the consciousness/emotion may adjust the knowledge to allow the obtained data for understanding, i.e., identifying the gap between the goal and current states as well as recognizing the constraints and opportunities imposed by the environment. In addition, when an action already taken is decided to be further improved, the consciousness/emotion may fix its knowledge to give a better plan later on. The structure of learning mechanism is also shown in Figure 1 with dotted-line connection.

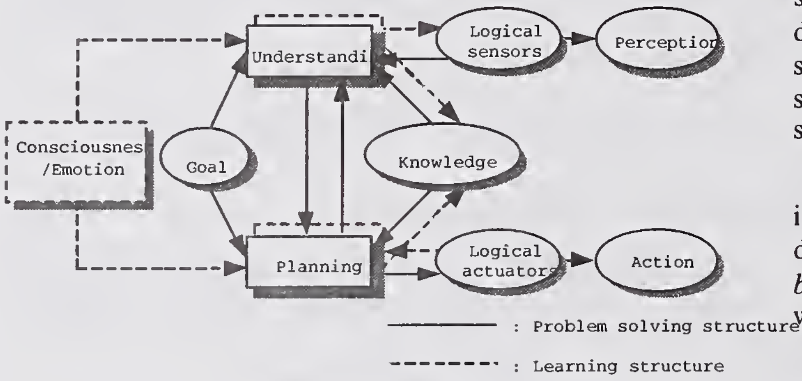


Figure 1. Structure of Machine Intelligence

The logical sensors and actuators as well as knowledge and constraint can be represented by an equivalent linguistic form. The same is true for representing the connection and projection associated with understanding and planning. If the functions of a system embedded in its hardware and software can be represented as a linguistic equivalent, based on the above observation, the MIQ and dMIQ of the system may be defined in the equivalent linguistic space. Thus, for a given machine to measure its intelligence, transforming the machine itself into this structure of problem solving and learning is first conducted, and then transforming it into the equivalent linguistic structure is to be done, which is discussed in the next section.

4. MEASURE BY LINGUISTIC EQUIVALENCE

Transforming system architecture into an equivalent formal language structure, a consistent measure of machine intelligence associated with the corresponding formal language can be obtained.

Any generic language used to build models representing diverse architectures must contain mechanisms to implement the features of all these architectures. For example, the parallel structure of the subsumption model requires parallelism in the language. At the other extreme, the functionality of a centralized planner must also be representable. If the structure of the model differs, we must be prepared to clearly determine equivalent operation.

4.1 The Machine Description Language

The basic unit of the Machine Description Language (MDL) is a behavior. The behavior nit is analogous to a sentence or statement constructed according to grammatical rules. There statements are conglomerated to form a meaningful system. The paper defines the grammatical rules of syntax of the Machine Description Language. Generating the semantics of an entire system is analogous to writing a program in a given system.

An MDL model has a hierarchical layered architecture composed of a number of various behaviors, some simple, and some complex. The simplest possible behavior is based on direct triggering by a single binary sensor which elicits a simple actuator response. For example, an on/off contact switch can trigger a behavior called "bump" which causes a short reverse movement combined with a turn.

Behavior modules are collected in groups which implement a complete autonomous task, such as obstacle detection. The collection of behaviors is called a *wrapped behavior.* The linguistic analogy is a paragraph of subroutine which encapsulates a single topic or function.

The composite wrapped behavior collectively implements some useful autonomous task. For example, a group of bump behaviors based on different contact sensors can be wrapped to form an obstacle rerouting wrapped behavior based on direct contact. If ultrasonic range detectors are added, new strands can be added to the composite object rerouting behavior, and the improved behavior them before bumping them. The old bump behaviors are kept as backups.

4.2 Analytical Measures with MDL

The performance of the system described here can be measured using traditional back box empirical techniques. For example, we can time its performance in executing a prescribed task. Alternatively, structural (linguistic) analyses of the system can be used to determine theoretical bounds on performance independent of implementational efficiency.

Structural analysis begins with identification of measurable quantities and their effects on performance. Many structural features can be measured; each contributes to the emergent intelligence of the completed system in a different way.

4.2.1 Behavior Attributes

We first consider measurable attributes of a behavior. Some of the measurable structural features are:

Strand Count and Strand Segment Count: A behavior has some number of strands (i.e., sensor to actuator information path) associated with it. Strands are regarded as instantaneous communication links for the purpose of measurement. The information packet propagation time between nodes, trigger, and taps is zero. The number and thickness of strands in a single behavior provides a measure of the resolution of sensory information, trigger situation discernability, and the dexterity or controllability of the actuator system. More fundamentally, strand *segment* count and thickness together measure the information transport capacity of the behavior.

Node Count: Node count captures the complexity of the sensor and actuator trees of a behavior. The node count is taken as the sum of nodes and taps for both sensor and actuator trees.

Trigger Propagation Time: Each trigger has three measurable attributes indicating the dimensionality of the input (parameters of the sensed situation), the dimensionality of the output (parameters of the desired response, based on the sensed situation) and the propagation time of the information, i.e., the delay between a sensed situation and the resultant response.

Node Propagation Time: The delay an information packet encounters between the time it enters a tap node, fusion node, or arbitration node and the time it (or the effects of a change in the information) exits the node, is termed *node propagation time*. It represents the processing time required to fuse information, to arbitrate competing controls, or to extract or combine information.

Strand Propagation time: The strand propagation time id the time for an information packet to travel from the sensor at the beginning of the strand to the actuator at the end of the strand.

Behavior Response Time: The response time of a behavior is the sum of all information propagation timers along the longest path between raw sensor input and raw actuator output. The path may include nodes from other behaviors but will include only one trigger propagation time. This differ from the propagation time of the longest strand in that the strand propagation time is measured from tap to tap, whereas the behavior response time is measured from raw sensor input to raw actuator output. Behavior response time is computed as:

$$\boldsymbol{B} = \max_i \left(\sum \alpha_i + \tau \right) \tag{4}$$

where

$\boldsymbol{B}$: behavior response time

α_i : node propagation time for node i

τ : trigger propagation time

Behavior response time can also be measured empirically, as long as the response can be isolated from the response of all other behaviors.

4.2.2 System Attributes

Next we consider attributes of the combined system:

Trigger of Behavior Count: The number of separate triggers (which is equivalent to the number of behavior modules) indicates the number of separate situations and corresponding responses, which the system can elicit, based on its sensory information. The total number of triggers in the entire system is and indication of complexity of the system and sophistication of response (assuming a well-designed system).

Strand Distribution: Strands which rely on many lower level strands provide more abstract, goal-directed, and strategic stimulus-response relationships, whereas the lower level strands provide greater reactivity and quicker response. The distribution of the strands between these extremes indicates the tendency for the system to generate behavior based on reflexes or impulses vs. goal-seeking behavior. One measure of this characteristic is the distribution of behavior propagation times. Standard statistical measures such as mean and median behavior propagation times, standard deviation, minimum and maximum propagation, describe the distribution. A median propagation time biased toward the minimum indicates a more quickly responsive and reflexive system whereas a bias toward the maximum indicated a deliberative system.

Layering Depth: Another measure of deliberativeness is the layering depth. The layering depth can be measured as the number of trees belonging to different behaviors which an information packet must traverse to reach the raw motors from the trigger. Because each group of wrapped behaviors comprises an autonomous set of behaviors, the layering depth or maximum depth of wrappers indicated the sophistication of autonomy. A system, which is more deeply wrapped, may indicate that it can perform more complex tasks autonomously. Each behavior added to a wrapped behavior indicates that some environmental situation can arise which s not handled optimally by the wrapped behavior by itself. If a wrapped behavior s itself wrapped along with new behaviors, the newly wrapped set handles all the environmental stimuli of the original wrapper plus all the situations detected by the new behviors.

MIQ: The *MIQ* (Machine Intelligence Quotient) is then defined as the product of the complexity of tasks the system can handle and the performance in task execution. This measure embodies the tradeoff between reflexivity (speed) and deliberativity (complexity). Task complexity is dependent both on the complexity and quantity of the tree structures. The complexity of tasks can be measured using the system attributes listed above, namely, trigger count, strand distribution, layering depth, strand count, and node count. We combine these as a weighted sum:

$$\boldsymbol{T} = w_\gamma \gamma + w_\delta \delta + w_\lambda \lambda + w_\sigma \sigma + w_\kappa \kappa \tag{5}$$

where

$\boldsymbol{T}$: Task complexity ability

γ : Trigger count

δ : Average strand propagation time overall machine

λ : Layering depth

σ : Total strand count in machine

κ : Total node count in machine

$w_\gamma, w_\delta, w_\lambda, w_\sigma, w_\kappa$: Respective weights

Performance in task execution is derived from the collective performance of behaviors. This can be computed as the weighted sum of behavior response time and *inverse* average strand propagation time (since speed increase as strand length decreases):

$$\boldsymbol{E} = w_B \boldsymbol{B} + w_\delta / \delta \qquad (6)$$

MIQ is then

$$MIQ = \boldsymbol{T} \cdot \boldsymbol{E} \qquad (7)$$

Resource: Machine "resource" is a measure of implementation requirements based on the architectural design of the machine. The resource is defined as the product of cost and volume. We compute the resource based on the number of processors and communication links required to implement the system directly in a parallel architecture. Processors are expensive while communication links are cheap. However, communication links can become numerous and occupy a large part of the volume of a machine. These costs and volumes are likely to change with new technology. The cost of the system is the sum of the costs of the processors (trigger, nodes, and taps) required. We assume one simple processor per trigger, node, or tap. We denote this as

$$\boldsymbol{C} = C_\gamma \gamma + C_\pi \pi \qquad (8)$$

where

$\boldsymbol{C}$: cost of machine

π : node count

C_γ, C_π : cost of trigger and node processors

The volume of the system is computed the same way:

$$\boldsymbol{V} = V_\gamma \gamma + V_\pi \pi \qquad (9)$$

Then resource is

$$\boldsymbol{R} = CV \qquad (10)$$

and the *rMIQ* is

$$rMIQ = MIQ / \boldsymbol{R} \qquad (11)$$

5. ENGINEERING CASE STUDY

A simple grasp controller based on the subsumption style of robot control uses a gripper beam and finger contacts as sensors as shown in Figure 2.

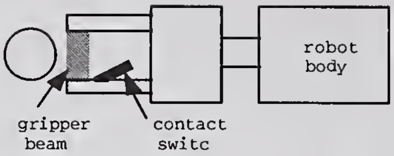


Figure 2. A Simple Robot Arm

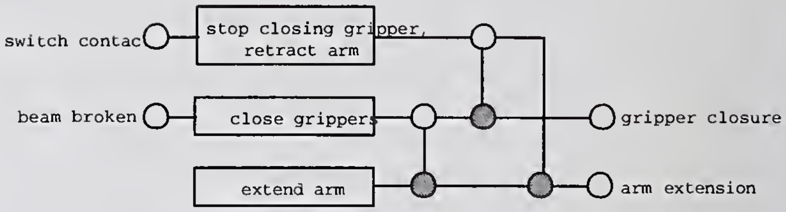


Figure 3. Subsumption Network

Figure 3 illustrates the simple subsumption network which generates the behavior of the robot. The extend arm behavior is always extends the arm (we ignore the condition of a fully extended arm). As soon as the gripper beam is broken, the sensor causes the "close grippers" behavior to trigger. The white motor node simultaneously inhibits the arm from extending with an inhibition node and activates the gripper closure actuator, causing the gripper to begin closing. (The gray nodes are taps – in this example they are motor taps or arbitrators.) When the grippers contact the object, the contact switch is closed, causing the "stop closing gripper, retract arm" behavior to trigger. The white node on the output of this behavior is a sequential node – first the gripper closure motor strand is inhibited, causing the gripper to first stop squeezing. Finally, the behavior subsumes the output of the "extend arm" behavior using a subsumption node, causing the arm to retract.

The *MIQ* and *dMIQ* of this system is easy to compute. All weights are set to one to simplify the example. There are three behaviors. The "extend arm" behavior is a trigger and a raw motor node (the tap nodes belong to the /"close gripper" and "stop gripper..." behaviors). The behavior response time for "extend arm" is therefore 1 + 1 = 2. There is one strand in this behavior. The "close grippers" behavior has one raw sensor node, one motor node tree node, and either one raw motor node or one motor tap; both of the two strands are the same length, so we may use either. The response time is 3 + 1 = 4. The "stop closing..." behavior similarly has a response time of 4 and a strand count of two. The mean behavior response or propagation time is (2 + 4 + 4) / 3, or 3.333. Layering depth is two, and system strand count is 5. Average strand propagation time over the entire system is (3 + 3 +3 + 3 + 1) / 5, or 2.6. There are nine nodes and nine strand segments in the entire system.

Based on these numbers, task complexity ability is 3 + 2.6 + 2 + 5 + 9 = 21.6. Remember, this number means little except as a comparative measure. Performance is 3.333 + 0.385 = 3.718. MIQ is then roughly 21.6 + 3.7 = 25.3. If we assume costs and volume of one, then cost and volume are both 9 + 9 = 18. Resource is (18)(18) = 324, and the rMIQ is 21.6/324 = 0.0667

6. CONCLUSION

We have presented three important issues, which should be considered when measuring machine intelligence, and introduced the structure of machine intelligence, which shows the internal mechanism of machine taking into account the three issues. Any machine can be represented by the proposed structure and the structure can be transformed into an equivalent linguistic structure so that one may define the metric of the machine intelligence in an analytical way.

In this paper, an equivalent linguistic structure has been proposed. It needs to be further developed to present linguistic structure of machine intelligence for both *MIQ* and *dMIQ* with respect to goals and time.

The formulation on *MIQ*, *dMIQ*, and *rMIQ* in Section 2 will be a good guide for defining machine intelligence since its clearness in the sense of goal-dependency, time-varyingness, and resource-dependency.

REFERENCES

[1] Albus, J. S., "Outline for a Theory of Intelligence," *IEEE Tran. on Systems, Man, and Cybernetics*, Vol. 21, No. 3, 1991, pp. 473-509

[2] Antsaklis, P. J. "At the gates of the Millennium: Are we in control?," *IEEE Control Systems Magazine*, Vol. 20, No. 1, February 2000, pp. 50-55

[3] Chalfant, E. C., and Lee, S., "Measuring the Intelligence of Robotic Systems: An Engineering Perspective," *Proceedings of International Symposium on Intelligent Systems*, Gaithersburg, MD, October, 1999

[4] Bien, Z., Kim, Y.-T., and Yang, S., "How To Measure The Machine Intelligence Quotient (MIQ): Two Methods And Applications," *Proceedings of World Automation Congress*, Alaska, May 9-14, 1998

[5] Bien, Z., Bang, W.-C., and Han, J.-S., "A Perspective of Intelligent Machine and Measurement of Machine Intelligence Quotient," *Proceedings of the 4th Korea-Japan Joint Workshop on Advanced Teleautomation and Intelligent Mechatronics*, Taejon, Korea, Nov. 4-6, 1998, pp. 64-67

[6] Lee, S., Schenker P. S., and Park, J., "Sensor-Knowledge-Command Fusion Paradigm for Man/Machine Systems," *Proceedings of SPIE International Symposium on Advanced Intelligent Systems*, Boston, MA, 1990

Formal Specification of Performance Metrics for Intelligent Systems

Ying Zhang
System and Practice Lab, Xerox Palo Alto Research Center
Palo Alto, CA 94304
Alan K. Mackworth
Department of Computer Science, University of British Columbia
Vancouver, B.C., Canada, V6T 1Z4
Email: yzhang@parc.xerox.com, mack@cs.ubc.ca

ABSTRACT

There are now so many architectures for intelligent systems: deliberative planning vs. reactive acting, behavioral subsuming vs. hierarchical structuring, machine learning vs. logic reasoning, and symbolic representation vs. procedural knowledge. The arguments from all schools are all based on how natural systems (i.e., biologically inspired, from basic forms of life to high level intelligence) work by taking the parts that support their architectures. In this paper, we take an engineering point of view, i.e., by using requirements specification and system verification as the measurement tool. Since most intelligent systems are real-time dynamic systems (all lives are), requirements specification should be able to represent timed properties. We have developed timed ∀-automata that fit to this purpose. We will present this formal specification, examples for specifying requirements and a general procedure for verification.

KEYWORDS: *formal specification, constraint-based requirements, system verification*

1. INTRODUCTION AND MOTIVATION

Over the last half a century, intelligent systems have become more and more important to human society, from everyday life to exploration adventures. However, unlike most other engineering fields, there has been little effort towards developing sound and deep foundations for quantitatively measurement and understanding such systems. The lack of measurement and understanding leads to unsatisfactory behavior or even potential danger for customers. The systems may not achieve desired performance in certain environments, or, the systems may even result in catastrophe in life-critical circumstances.

Many researchers have suggested measures of performance for intelligent systems, such as the Turing Test [12], Newell's expanded list [9,10] and Albus's definition of intelligence [4]. However, most of these measures are not based on formal quantitative metrics. There are also efforts on comparing performance on pre-defined tasks, such as a soccer competition [11]. However, these methods are domain specific therefore hard to apply to general cases. We advocate formal methods for specifying performance requirements of intelligent systems. Much research has been done on formal methods (http://archive.comlab.ox.ac.uk/formal-methods.html) over the last twenty years. In this paper, we explore one of the approaches, namely, using timed ∀-automata for specifying performance requirements.

The timed ∀-automata model was developed in [13, 17] as an extension of discrete time ∀-automata [8] to continuous time, annotations with real-time. Timed ∀-automata are simple yet able to represent many important features of dynamic systems such as safety, stability, reachability and real-time response. In the rest of this paper, we introduce the formal definition of timed ∀-automata first, then present examples of timed ∀-automata for representing performance metrics, and finally describe a general verification procedure for this type of requirements specification.

2. TIMED ∀-AUTOMATA

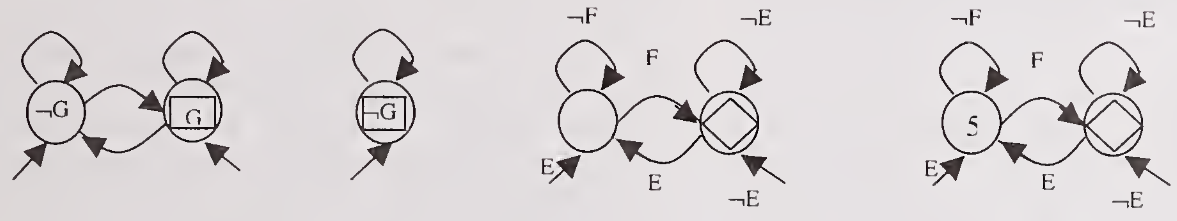
In general, there are two uses of automata: 1. to describe computations, such as input/output state automata, and 2. to characterize a set of sequences, such as regular grammars/languages. Examples of the first category are mostly deterministic and examples of the second category are mostly non-deterministic. However, all the original automata work is based on discrete time steps/sequences. Approaches to extending automata to continuous time have been explored in hybrid systems community over the last decades [1,2,7]. The timed ∀-automata model that we developed belongs to the second category, i.e., *non*-deterministic *finite* state automata specifying behaviors over *continuous* time. The discrete time version of ∀-automata was originally proposed as formalism for the specification and verification of temporal properties of concurrent programs [8].

2.1. *Syntax*

Syntactically, a timed ∀-automaton is defined as follows.

[**Definition 1**] A *∀-automaton* A is a quintuple (Q, R, S, e, c) where Q is a finite set of automaton-states, $R \subseteq Q$ is a set of recurrent states and $S \subseteq Q$ is a set of stable states. With each $q \in Q$, we associate an assertion e(q), which characterizes the entry condition under which the automaton may start its activity in q. With each pair q, q' $\in$ Q, we associate an assertion c(q, q'), which characterizes the transition condition under which the automaton may move from q to q'. R and S are generalizations of accepting states. We denote by $B = Q - (R \cup S)$ the set of non-accepting (bad) states. Let R^+ be the set of non-negative real numbers representing time durations. A *timed ∀-automaton* is a triple (A, T, τ) where A is a ∀-automaton, $T \subseteq Q$ is a set of timed automaton-states and $\tau: T \cup \{B\} \rightarrow R^+ \cup \{\infty\}$ is a time function.

One of the engineering advantages of using automata as a specification language is its graphical representation. It is useful and illuminating to represent timed ∀-automata by diagrams. A timed ∀-automaton can be depicted by a labeled directed graph, where automaton-states are depicted by circle nodes and transition relations are by directional arcs. In addition, each automaton-state may have an entry arc pointing to it; each recurrent state is depicted by a diamond and each stable state is depicted by a square, inscribed within a circle. Nodes and arcs are labeled by assertions as follows. A node or an arc that is left unlabeled is considered to be labeled with **true**. Furthermore, (1) if an automaton-state q is labeled by ψ and its entry arc is labeled by φ, the entry condition e(q) is given by $e(q) = \psi \wedge \varphi$; if there is no entry arc, e(q) = **false**, and (2) if arcs from q to q' are labeled by φ_i, i = 1...n, and q' is labeled by ψ, the transition condition c(q, q') is given by $c(q, q') = (\varphi_1 \vee \ldots \vee \varphi_n) \wedge \psi$; if there is no arc from q to q', c(q, q') = **false**. A T-state is denoted by a nonnegative real number indicating its time bound. Some examples of timed ∀-automata are shown in Figure 1.



(a) (b) (c) (d)

Figure 1. Examples of timed ∀-automata

2.2. *Semantics*

Semantically, each assertion denotes a constraint defined on a domain of interest. Let D be a domain of interest; D can be finite, discrete, or continuous, or a cross product of a finite number of domains. Physically, D can represent, for example, speeds, distances, torques, sentences, commands or a combination of the above. A constraint C defined on D is a subset of D, $C \subseteq D$. Physically, a constraint represents certain relation on a domain, such as a relation between external environment stimuli and an agent's internal knowledge representation, or, a relation between internal states and actions, or, the relation between the current and next state. An element d in domain D satisfies constraint C, if and only if $d \in C$.

The semantics of timed ∀-automaton is defined as follows. Let *T* be a time domain, which can be continuous, for example, R^+. First, let us define runs of ∀-automata. Let A = (Q, R, S, e, c) be a ∀-automaton and $v: T \rightarrow D$ be a function of time. A *run* of A over v is a function $r: T \rightarrow Q$ satisfying:

1. *Initiality*: $v(0) \in e(r(0))$;
2. *Consecution*:
   a. *Inductivity*: $\forall t>0$, $\exists q \in Q$, $t'<t, \forall t''$, $t' \le t'' < t$, $r(t'')=q$ and $v(t) \in c(r(t''), r(t))$ and
   b. *Continuity*: $\forall t$, $\exists q \in Q$, $t'>t$, $\forall t''$, $t<t''<t'$, $r(t'')=q$ and $v(t'') \in c(r(t), r(t''))$.

   When *T* is discrete, the two conditions in *Consecution* reduce to one, i.e., $\forall t>0$, $v(t) \in c(r(pre(t)), r(t))$ where pre(t) is the previous time point of t.

If r is a run, let Inf(r) be the set of automaton-states appearing infinitely many times in r, i.e., $\mathrm{Inf}(r) = \{q | \forall t \exists t' \ge t, r(t')=q\}$. A run is called *accepting* if and only if

1. $\mathrm{Inf}(r) \cap R \neq 0$, i.e., some of states appearing infinitely many times in r belong to R, or
2. $\mathrm{Inf}(r) \subseteq S$, i.e., all the states appearing infinitely many times in r belong to S.

For a timed ∀-automaton, in addition for a run to be accepting, it has to satisfy time constraints. Let $I \subseteq T$ be a time interval and |I| be the time measurement, and let r|I be a segment of r over time interval I. A run satisfies time constraints if and only if:

1. *Local*: For any $q \in T$ any time interval I, if $r|I$ is a segment of consecutive states of q, then $|I| \leq \tau(q)$;
2. *Global*: For any time interval I, if $r|I$ is a segment of consecutive states of $B \cup S$, then $\int_I \chi_B(r(t))dt \leq \tau(B)$, where $\chi_{B:} Q \rightarrow \{0,1\}$ is the characterization function for the set B.

[Definition 2] A timed ∀-automaton TA = (A, T, τ) accepts a trace v, if and only if

1. All runs are accepting for A;
2. All runs satisfy the time constraints.

With the semantics defined, we can infer that, for the timed ∀-automata in Figure 1, (a) specifies the behavior of reachability, i.e., eventually the system should satisfy constraint G, (b) specifies the behavior of safety, i.e. constraint G is never satisfied, (c) specifies the behavior of bounded response, i.e., whenever constraint E is satisfied, constraint F will be satisfied within bounded time and (d) specifies the behavior of real-time response, i.e., whenever constraint E is satisfied, constraint F will be satisfied within 5 time units.

3. EXAMPLES OF PERFORMANCE SPECIFICATION

Timed ∀-automata are simple yet powerful for the specification of behaviors of dynamic systems, since it integrates constraint specification with timed dynamic behavior specification.

3.1. *Examples of Constraint Specification*

Constraint specification alone can specify many performance metrics. Constraints specify relations between external environment stimuli and an agent's internal knowledge representation, or between internal states and actions, or between the current and next states. Constraints can be finite, discrete or continuous, or any combination of the above. Constraints can be linear, nonlinear, equalities or inequalities. Moreover, constraints can also specify optimal conditions or optimality with extra constraints, or combinations of multiple optimal criteria and additional constraints.

Considering the following examples for specifying constraints:

1. *Inequality:* $f(x) \leq 0$ where x is a vector of variables and f is a vector of functions.
2. *Optimality*: min $|f(x)|$ where $|x|$ is a norm for x.
3. *Negation*: $x \neq y$.
4. *Constrained Optimality*: min$|f(x)|$ given $g(x) \leq 0$.
5. *Robustness*: Let f(x) be a set of output functions with x as inputs. The robustness can be represented by its Jacobian $J = \Delta f/\Delta x$. There are many ways to state an optimal condition for robustness. One method is to minimize $|w|$ where w is the diagonal elements of W in the singular value decomposition of $J = UWV^T$.

3.2. *Examples of ∀-Automata*

With automata, timed dynamic behaviors can be specified. Here is a set of examples for specifying performance using timed ∀-automata, as shown in Figure 1:

1. Let G be a constraint that the distance between the robot and its desired position is less than some constant value. Then Figure 1(a) specifies that the robot will eventually arrive its desired position.
2. Let G be a constraint that the error of a learning algorithm is less than a desired tolerance. Then Figure 1(a) specifies that the learning will eventually convergence. If let the state of ¬G in Figure 1(a) as a timed state with time bound t, it further specifies that the learning will be done within time t.
3. Let G be a constraint that the distance between the robot and obstacles is less than some constant value. Then Figure 1(b) specifies that the robot will never hit any obstacle. If G denotes that the current memory usage is out of the limit, Figure 1(b) specifies that the memory usage at any time is within its limit.
4. Let E be an external stimuli and F be a response. Then Figure 1(c) specifies that there is a response after stimuli within bounded time. Figure 1(d) specifies that such a response is within 5 time units.

Even though timed ∀-automata are powerful, still they are not able to represent all forms of performance metrics. For example, optimal performance over time min$\int f(t)dt$ is not specifiable with timed ∀-automata. This form is mostly used for characterizing energy, efficiency or overall errors. Furthermore, specification with probability behaviors are not included either. However, it is not hard to add probability, for example, instead of "all runs" must be accepting and satisfying time constraints, we can say "x% runs" must be accepting and satisfying time constraints.

3.3 *Performance Comparisons*

Note that requirements specification defines what the system should do, rather than defining how the system is organized, i.e., its architecture. For example, behavior-based control [4,6] (which is arbitration based or a horizontal hierarchy) has a different form of architecture from function-based control [5] (which is abstraction-based or a vertical hierarchy); model-based systems have a different form of architecture from learning-based systems, event-driven systems have a different kind of architecture from time-driven systems. Different systems with different architectures can still be compared based on the behavioral interface under the formal performance specification. For example, given a set of requirements specification Rs and system A satisfies a subset As $\subseteq$ Rs and system B satisfies a subset Bs $\subseteq$ Rs. If As $\subseteq$ Bs, system A is not better than system B with respect to requirements Rs. Similarly, if system A satisfies requirement α and system B satisfies requirement β and if α implies β, system A is better than system B with respect to the requirement.

However, this specification does not define metrics on architectures. The measurement of performance should come from the customer's point of view, but the measurement of architecture should come from the developer's point of view, i.e., design time, debug time, upgrading time, modularity and the percentage of re-usable components.

4. SYSTEM VERIFICATION

For most dynamic systems, stability or convergence is the most important property that needs to be verified. For example, we can verify that equation $dx/dt = 0$ satisfies the property of $\forall$-automaton in Figure 1(a) with G as $|x|\leq\varepsilon$ for any positive number ε. The most commonly used method for the verification of such properties is the use of Liaponov functions. We developed a formal method based on model-checking, that generalizes Liaponov functions [13,17]. This method is automatic if the domain of interest is finite discrete and time is discrete [13].

The details of the model-checking method are out of the scope of this paper. The basic principle is to first find a set of invariants, each associated with an automaton-state in the timed $\forall$-automaton. Then, find a set of Liaponov functions, which are non-increasing in stable states and decreasing in bad states. Finally, find a set of local and global timing functions, where local timing functions are decreasing in timed states and global timing functions, like Liaponov functions, are non-increasing in stable states and decreasing in bad states, in addition to be bounded in values.

5. RELATED WORK AND CONCLUSION

Much work has been done in formal approaches to system specification and verification [1,2,7,8]. In general, there are two schools. One is to develop a uniform specification for both systems and their requirements; the other is to use two different specifications, one for systems and one for requirements. The advantage of the former is that the same formal approach can apply to both system synthesis and system verification. However, in most cases, if the specification language is powerful for both systems and requirements, the synthesis or verification tasks become hard. We advocate the latter approach, i.e., using timed $\forall$-automata for requirements specification and using Constraint Nets [13,18,19] for system modeling. Control synthesis [13,14] and verification [13,15,16,17,20] are also studied in this framework.

In this paper, we have shown how to use formal methods to specify the performance metrics of intelligent systems, with timed $\forall$-automata as an example. The advantage of formal methods over other methods lies in their precision and generality. Timed $\forall$-automata, with its graphical depiction and constraint specification, is a simple yet powerful formalism for specifying many properties of dynamic systems.

6. REFERENCES

[1] Alur, R., C. Courcoubetis, T.A Henzinger and P. Ho, "Hybrid automata: Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems," R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, editors, *Hybrid Systems*, LNCS 736, Springer-Verlag, 1993, pp. 209 – 229.

[2] Alur, R. and D. Dill, "Automata for modeling real-time systems," M.S. Peterson, editor, ICALP90: *Automata, Languages and Programming*, LNCS 443, Springer-Verlag, 1990, pp. 322 – 335.

[4] Albus, J.S., "Outline for a Theory of Intelligence," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 21, No. 3, pp. 473 – 509, May/June 1991.

[5] Arkin, R.C., *Behavior-Based Robotics*, The MIT Press, Cambridge, MA, 1998.

[6] Brooks, R.A., "Intelligence without reason," in IJCAI 1991, Sydney, Australia, pp. 569 – 595.

[7] Henzinger, T.A., Z. Manna and A. Pnueli, "Timed transition systems," J.W. deBakker, C. Huizing, W.P. dePoever, and G. Rozenberg, editors, *Real-Time: Theory in Practice*, LNCS 600, Springer-Verlag, 1991, pp. 226 – 251.

[8] Manna, Z. and A. Pnueli, "Specification and verification of concurrent programs by $\forall$-automata," in Proc. 14th Ann. ACM Symp. On Principles of Programming Languages, 1987, pp. 1-12.

[9] Newell, A., "The Knowledge Level," *Artificial Intelligence*, 18(1), pp. 87-127, 1982.

[10] Newell, A., and Simon, H., GPS: A Program that Simulates Human Thought," Feigenbaum and Feldman,

editors, *Computers and Thought*, McGraw-Hill, New York, 1963.

[11] Sahota, M. and A. K. Mackworth, "Can situated robots play soccer?" in Proc. Artificial Intelligence, 1994, Banff, Alberta, pp. 249 – 254.

[12] Turing, A. "Computing Machinery and Intelligence." *Mind 59*, pp. 433-460, 1950. Reprinted in Feigenbaum and Feldman, editors, *Computers and Thought*, McGraw-Hill, New York, 1963.

[13] Zhang, Y., "A Foundation for the Design and Analysis of Robotic Systems and Behaviors", PhD Thesis, University of British Columbia, Canada, 1994.

[14] Zhang, Y. and A. K. Mackworth, "Synthesis of Hybrid Constraint-Based Controllers," P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, editors, *Hybrid Systems and Automatic Control*, LNCS 999, Springer-Verlag, 1994, pp. 552 – 567.

[15] Zhang, Y. and A. K. Mackworth "Specification and Verification of Constraint-Based Dynamic Systems," A. Borning, editor, *Principles and Practice of Constraint Programming*, LNCS 874, Springer-Verlag, 1994, pp. 229 – 242.

[16] Zhang, Y. and A. K. Mackworth, "Will The Robot Do The Right Thing," in Proc. Artificial Intelligence, 1994, Banff, Alberta, pp. 255 – 262.

[17] Zhang, Y. and A. K. Mackworth, "Specification and Verification of Hybrid Dynamic Systems Using Timed ∀-Automata," *Verification and Control of Hybrid Systems*, LNCS 1066, Springer-Verlag, 1995.

[18] Zhang, Y. and A. K. Mackworth, "Constraint Programming in Constraint Nets", V. Saraswat and P. Van Hentenryck, editors, *Principles and Practice of Constraint Programming*, MIT Press, Cambridge, MA, 1995, pp. 49 – 68.

[19] Zhang, Y. and A. K. Mackworth, "Constraint Nets: A Semantic Model for Hybrid Dynamic Systems," *Journal of Theoretical Computer Science*, Vol. 138, No. 1, pp. 211 – 239, 1995.

[20] Zhang, Y. and Alan K. Mackworth, "Modeling and Analysis of Hybrid Systems: An Elevator Case Study," H.Levesque and F.Pirri, editors, *Logic Foundations for Cognitive Agents*, Springer, Berlin, 1999, pp. 370-396.

Metrics for Intelligence: the Perspective from Software Agents
PRELIMINARY NOTES

Line Pouchard
Collaborative Technologies Research Center
Computer Science and Mathematics Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831-6414
pouchardlc@ornl.gov

Each scientific development that claims to provide a "new way" for approaching existing problems needs proper (i.e. formal and quantifiable) evaluation methods and consensus-based criteria for measuring the validity of its claims. Taken together, these methods and criteria constitute the metrics by which new developments are being measured against their claims. Various claims have been made in the literature for the technology of intelligent software agents. Such claims include a new approach to programming providing a breakthrough comparable to the one achieved through object-oriented methods; an approach to programming that is more readily understood by non-programmers; an approach that lowers the costs of software inter-operability.

Software agents need proper metrics if the technology is to fulfill its promises and make a lasting impact. One characteristic distinguishing software agents from software developed with object-oriented and procedural methodologies is the anthropomorphic characteristics that agents exhibit. Various taxonomies for software agents currently exist [1, 2, 3]. Agents typically present one or several of the following characteristics:

- Pro-activeness and goal-orientation
- Reactiveness (reactive agents)
- Autonomy (rational agents, and others)
- Mobility (mobile agents)
- Learning and reasoning ability (deliberative agents, and others)
- Social ability: communication and cooperation (multi-agent systems)

An agent is considered intelligent if it can learn from its environment and modify its behaviors and goals to respond to environmental constraints that were uncertain and unforeseen at the time of development. Agents are thus particularly adapted to model environments where software components act autonomously on users' behalf and problem-solving environments where parameters of computation dynamically change during processing. The ability to learn for an agent is coupled with the ability to perform resource and knowledge discovery. This action may take the form of querying and updating knowledge-based systems. Knowledge discovery and interpretation bring latency to the agent and may impair the achievement of its overall goals. For instance, reactive agents that need a quick response time may not embody much learning and reasoning because the overhead renders the agent useless.

Software agents present one or some capabilities that are affected by the choice of specific components described in the Tools of Intelligence (see White paper). For instance, searching for a required object within a scene is one area where software agents have successfully been implemented. If you take the "scene" to be an information space like the Internet, information-gathering and retrieval agents display this capability and have been successful at performing the task. Deliberative agents such as Belief-Desire-Intention (BDI) agents exhibit the capability of remembering scenes and experiences as their Beliefs are based on this capability. These agents are also able to interpret and respond to unforeseen situations.

Agents' ability to autonomously execute processes on remote systems, given the appropriate permissions, is also a characteristic some intelligent systems (but not all) need to efficiently and effectively perform. This requires proper measures. This characteristic, known as mobility, has very different meaning for physical agents.

Mobility requires intelligence for software agents because true mobility requires resource discovery. For those agents designed as mobile agents the degree of mobility can constitute a measure of its intelligence. Mobile agents travel over networks such as the Internet and execute processes on remote platforms. Mobile agents may start execute a process on a particular machine, be unexpectedly interrupted, travel to another available platform, and continue the execution of the process from where it was interrupted. Such a mobile agent needs intelligence to interrupt and restart its execution autonomously without resetting, and for determining which resources to use in a networked environment. Network agents used for telecommunication applications (such as testing the reliability of a network) exemplify these types of agents.

Social intelligence needs to be measured in multi-agent systems. The degree of social interaction and the agents' ability to exhibit social behavior constitute an important criterion for multi-agent systems. Not all agent-based systems need to exhibit this characteristic (mobile agents may never need to talk to each other for instance). The type of social interaction between agents conditions knowledge acquisition and interpretation. The social model affects the individual pursuit of goals and may ultimately affect the survival of the system [4]. When one considers a multi-agent systems, there are at least two models. Both types of multi-agent systems, collaborative and cooperative, display the characteristics of open systems.

- Model 1: Each individual agent's goal is subservient to an over-arching goal of the system. We have a cooperative system, where agents agree not to pursue goals detrimental to each other and the whole system, even if these "careless" goals are in accordance with the individual agent's goal.

- Model 2: Each agent acts on its own behalf without recognizing a higher agent-entity with the ability to regulate its goals (there is still a need for a kind of supervisor agent that regulates communication). We have a collaborative system. This is the case for so-called rational agents, used especially in e-commerce, where agents act in a market-like environment, with the ability to bid for money on the goods and services each offers.

Agent-communication languages should theoretically let heterogeneous agents communicate, but none currently do [5]. A significant part of the inter-operability issue is the lack of a shared content language and ontology. An ontology expresses, for a particular domain, the set of terms, entities, objects, classes and the relationships between them with formal definitions and axioms that constraint the interpretation of these terms [6]. These definitions and axioms are written in a variety of logical languages (e.g. KIF [7]), and provide a formal theoretical basis to domain taxonomy. They can serve to automatically infer translation engines between software applications. By making explicit the implicit definitions and relations of classes, objects, and entities, ontologies also contribute to knowledge sharing and re-use across systems. The use of ontologies in agent-based systems is proposed as a criterion for the metrics of intelligent software agents. The degree of completeness and consistency of ontologies can be formally proven and provide a quantifiable criterion.

Ontologies constitute an important criterion for the metrics of intelligent software agents, in particular for agents exhibiting the social abilities of communication and cooperation. Software agents require the use of or a translation to a shared terminology and syntax in order to efficiently and effectively inter-operate. Agent-communication languages such as KQML meet the challenges of inter-operability with mitigated success [8]. Agent communication languages specify the possible use of ontologies in their syntax but do not require it. FIPA ACL proposes an ontology service as a normative specification [9].

In conclusion, software agents exist either as standalone or in social systems. Agents are made of components, and an agent-oriented architecture typically includes the agent application as well as an environment in which agents execute. They may execute on a single machine, on several machines connected locally or by wide-area network. These agents need a degree of mobility. They may be developed by different developers on different platforms, and therefore need a common communication language including protocol and ontologies (see [10] for an assessment of the state-of-the-art in this area). In addition, since agents may exhibit any combination of the characteristics above, some taxonomies of agents prefer a classification based on the domains in which software agents have been successfully implemented [11], rather than on their inherent characteristics.

Software agents also exist as whole, where an agent-based system is made of the agent and the underlying environment. The environment may include the knowledge repositories and ontologies which are key to the agents' degree of intelligence. For this reason, the mind/body dichotomy, and the proposition to measure the intelligence of the system based on the intelligence of the mind (controller), do not hold for agent based systems.

In addition to characteristics applicable to Constructed Systems with Autonomy, the metrics of intelligence for software agents need to include the following (not all these characteristics need apply for the same system):

- be domain-specific
- measure the degree of mobility
- present an agent communication language
- refer to ontologies.

References

1.Brenner, Walter; Zarnekow, Rudiger, and Wittig, Hartmut. *Intelligent Software Agents. Foundations and Applications*. Berlin: Springer Verlag; 1998; pp. 37-41.

2.Nwana, H. et. Al. "What is an agent?" Available at: http://www.labs.bt.com/projects/agents/publish/papers/review2.htm#agent

3.Wooldridge, M. Intelligent Agents. In *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*. Weiss, Gehrard, ed. Cambridge, Mass.: MIT Press; 1999; pp.27-73.

4.Huhns, Michael N. and Stephens, Larry. Multiagent Systems and Societies of Agents. In *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*. Weiss, Gehrard, ed. Cambridge, Mass.: MIT Press; 1999; pp. 79-120.

5.Singh, Munindar P. Agent communications languages: rethinking the principles. *IEEE Computer*. 1998; 31(12):40-47.

6.Gómez-Pérez, Asuncion. Knowledge Sharing and Reuse. In *The Handbook of Expert Systems*. Boca Raton, FL: CRC Press; 1998; pp. 10/1-10/36.

7.Knowledge Interchange Format. Available at http://logic.stanford.edu/kif/dpans.html.

8.Labrou, Yannis; Finin, Tim, and Peng, Yun. Agent communication languages: the current landscape. *IEEE Intelligent Systems & Their Applications*. 1999; 14(2):45-52.

9. Foundation for Physical Intelligent Agents (FIPA). Available at http://drogo.cselt.it/fipa and http://www.fipa.org.

10. Wooldridge, M. and Jennings, N.R. Applications of Intelligent Agents. In *Agent Technology: Foundations, Applications, and Markets*. Berlin: Springer Verlag, 1998; pp. 3-28.

11. Nwana, Hyacinth S. and Ndumu, Divine T. A perspective on software agents research. *The Knowledge Engineering Review*. 1999; 14(2):125-142.

Minimal Representation Size Metrics for Intelligent Robotic Systems

Dr. Arthur C. Sanderson
Department of Electrical, Computer and Systems Engineering
Rensselaer Polytechnic Institute
Troy, NY 12180
sandea@rpi.edu

ABSTRACT

The minimal representation size criterion provides a metric for the configurational complexity of robotic tasks and may be used to evaluate alternative algorithms, strategies, and architectures for the accomplishment of specific tasks. The principles of explicit and implict representation are used to define this complexity and the resulting information measure derived may be considered as a measure of configurational intelligence of the system. Specifically, these measures indicate the internal explicit information required to specify the accessible states of the robotic system using its available perception and actuation capabilities. The resulting approach may be used to evaluate and guide applications tasks such as robotic assembly and multisensor manipulation.

Keywords: *minimal representation size, intelligent systems, performance metrics, robotics*

1. INTRODUCTION

Intelligent robotic systems couple computational intelligence to the physical world and such systems may be considered as intelligent agents that perceive the environment, and select an action or sequence of actions to affect the environment. Such an intelligent agent constructs an internal "representation" of the environment, and uses reasoning to choose among alternative actions. Specifically, we can define robots as "active, artificial, intelligent agents whose environment is the physical world". Such agents may be distinguished from software agents, human agents, and others.

Such an intelligent robot is regarded as "rational" if the agent makes decisions to choose actions that accomplish a known task goal, or increase a performance measure of the task. It is important to distinguish the presence of intelligence from the metric of performance. Intelligence (reasoning), in itself, does not maximize overall performance. However, intelligence may be used to choose among a set of candidate actions that may improve performance or achieve a goal.

An intelligent robot may also be characterized by its autonomy. In the context of these definitions, autonomy refers to the capacity of the robot to define its own goals or sub goals, often based on its perception and internal representation of the environment. Autonomy widens the scope of tasks, which the same system can perform without reprogramming, but in general, requires more sophistication in the design and architecture of the system. The non-autonomous system may accomplish a smaller set of tasks and may require efforts to constrain or redesign the environment to conform to task assumptions.

The structure of an intelligent robot agent includes perception, representation, reasoning, and representation. The implementation of such an agent requires two major components: (1) Algorithms that define the representation structure and reasoning sequence, and (2) Architecture that defines the organization of the system to accomplish set goals and performance. In practice, the selection of the architecture has been strongly intertwined with the nature of the representation. For example, one simple intelligent robot defines a perception-action pair such as "move hand if you touch the hot stove!" Such a reflex action might be expressed as a look-up table in which state representation is a simple binary element.

As the complexity of robots and tasks increases, a single reflex action is inadequate to create required behaviors, and architectural approaches have tended to evolve in two directions. First, *hierarchical* architectures have been based on the definition of a hierarchy of *explicit* representation of the robot state. A hierarchy of perceptual representation may involve image features, shapes, objects, scenes, etc., while a hierarchy of actions may involve joint motion, arm motion, robot motion, sensor-based motion etc. The formal definition of such a hierarchical architecture [1] has provided an important basis for building consistent, predictable, and programmable robotic systems.

A second trend has been the development of *behavioral* architectures [3] that expand upon simple reflexes by creating a network of interdependent reflexes in order to increase the sophistication of the behaviors. One such behavioral approach is the *subsumption* architecture [5] that utilizes finite state machines to impose a priority setting logic on the reflex actions. The nature of such behavioral architectures is to incorporate an *implicit* representation of the environment in order to define a simplified state space of perceptions and actions. From a systems perspective, the behavioral architecture utilizes constraints or assumptions about the environment to identify a subspace (manifold) within the explicit state space. A reflex action, or set of actions, may then be defined within the subspace with the logical consistency to achieve goals and performance metrics.

The distinction between *explicit* and *implicit* representations is important to the interpretation of intelligence in systems. A simple task example helps to illustrate these distinctions. Consider a room with a single door containing a mobile robot. The robot task goal is to exit the room, and it may have a performance metric of minimum time to exit. Several different types of algorithms may be considered:

(1). Random search (Figure 1a)
The robot moves in random directions without using perception, mechanically bouncing off the walls. Eventually, it is guaranteed to exit the room.

(2). Wall following – simple reflex (Figure 1b)
The robot uses a simple sensor to detect presence or absence of an adjacent wall. The algorithm:

```
    IF ('wall-is-in-front') THEN ('Turn-Right') ELSE
    ('Follow-wall-on-left')
```

is guaranteed to find the door, though the path may be long.

(3). Perception - Explicit state representation (Figure 1c)
The robot uses a sophisticated vision sensor to view the door, acquire a perception, P, update the global internal state representation, GS, and plan an explicit path to the door.

(4). Perception – Implicit state representation (Figure 1d)
The robot defines an implicit mapping of GS to local state, LS, that is consistent with the desired goal state. By mapping perception into LS, rather the GS, the resulting algorithm is often more efficient and simpler to implement. In this case, consider a sensor that perceives only the width, W, of the door, but no other attributes of the environment. We choose W to be the local state representation, LS = W, and define a local reflex algorithm to choose an action, A:

Choose A to increase W.
(a). If robot, R, moves toward the door, W' > W.
(b). If R moves perpendicular to the door, then W'>W.

The resulting *local* changes in W move the robot toward and through the door, achieving the global goal. However, LS is never sufficient to explicitly locate the robot in the room, i.e. determine GS. This strategy is analogous to a potential field mapping related to the perceived door width feature of the room. The same strategy may be used as a feature-based method to guide a peg-in-hole or other assembly problem using visual servoing of the area of the target hole [26].

These examples illustrate several types of tradeoffs in the design of intelligent systems, and also confirm that the most intelligent system may not result in the optimal performance on a given task, as illustrated in the performance of the feature-based example. First, for this purely geometric task, we can define one component of the intelligence of the system, the *configurational complexity* as the *information required to represent the accessible states of the internal representation of the system.* "Accessible states" are defined as those states that may be achieved as goal states of the system through its perception-action algorithms. In this sense, the representational intelligence of the system is equated to the size of the internal representation space.

For the examples in Figure (1), the configurational complexity is found to be: (a). 1 bit, (b). 3 bits, (c). 30 bits, and (d). 10 bits, where a resolution of 10 bits has been assumed for the vision sensor used in (c) and (d). By considering the approximate number of steps required to achieve the result, on can similarly compute the cumulative complexity for each of the tasks to be: (a). 100 bits, (b). 75 bits, (c). 60 bits, and (d). 20 bits. Therefore, the minimal complexity approach to the task is given by strategy (c) and may be regarded as a tradeoff between explicit and implicit information needed for the task.

In addition, the time (number of steps) required for each task is implicit in the cumulative information and reflects the inherent deficiencies in the worst case scenarios for (a) and (b). Based on the viewpoint of encoded residuals discussed in the next section, one can also calculate the encoded implicit information for each strategy: (a). 20 bits, (b). 18 bits, (c). 0 bits, (d). 12 bits.

Figures (e) and (f) emphasize the inherent assumptions that are often present in such systems. Strategies (a) and (b) are not guaranteed to succeed for problems (e) and (f), where the subspace manifold defined by the strategy is no longer guaranteed to contain the goal. Strategies (c) and (d) may still succeed but require more steps and a more sophisticated algorithm.

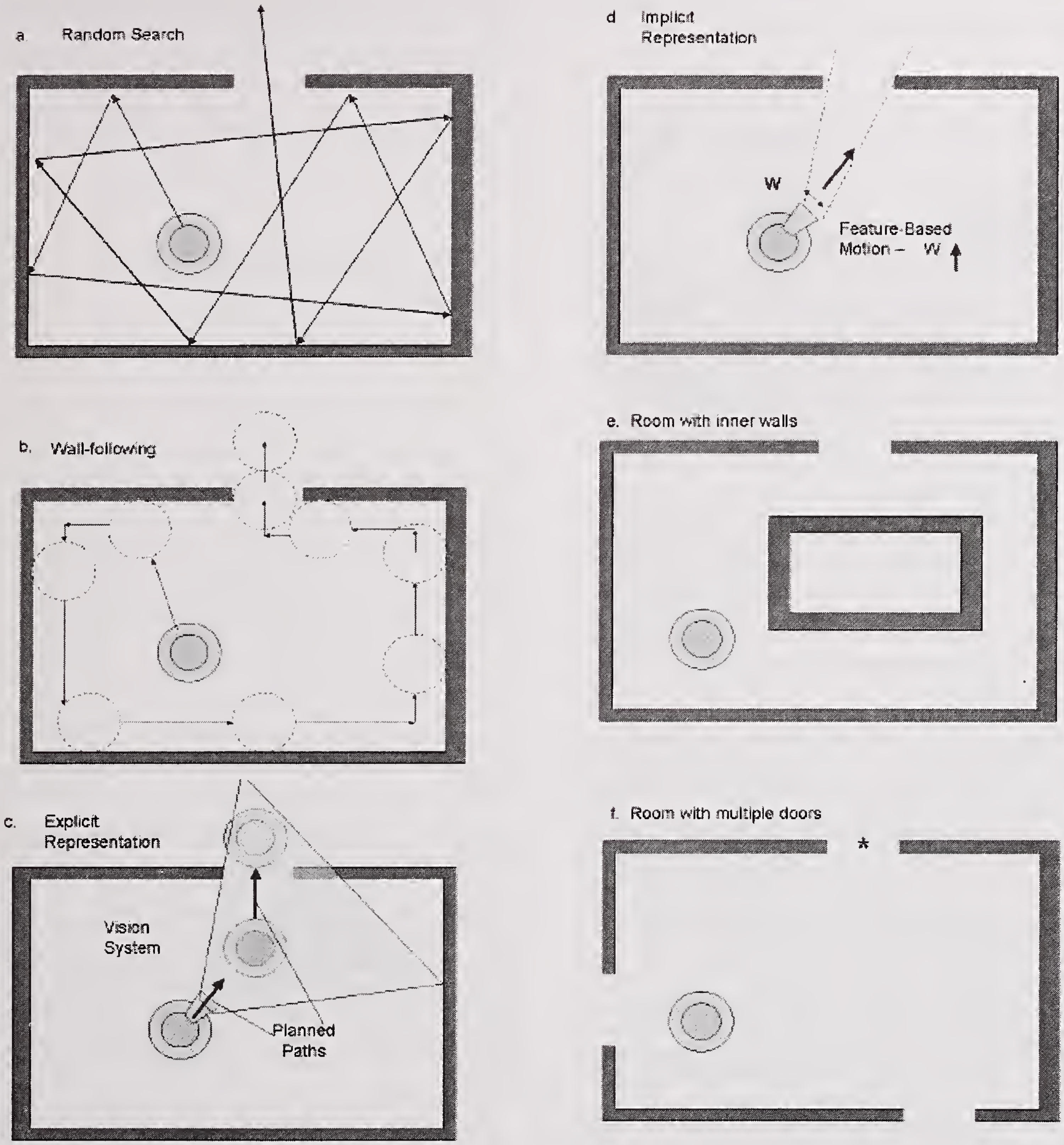


Figure (1). Examples of alternative strategies for the task of exiting a room through the door: (a). Random search, (b). Wall-following, (c). Explicit representation and global planning, (d). Implicit representation and local reasoning. All four strategies will accomplish the basic task. However, (a) and (b) are not general and will fail when the environment differs from the basic assumptions, such as in (e) with inner walls, and in (f) with multiple doorways.

2. MINIMAL REPRESENTATION SIZE

The minimal representation size (MRS) methods [6,18,19,23,24] used in this work are also called "minimum description length" methods in the literature. The MRS approach introduces an information measure of model complexity and has been applied to a number of related problems in attributed image matching [22], shape matching [11], density estimation [4], and model based sensor fusion [11-17]. The minimal representation criterion defines the minimal overall data representation among a choice of alternative models and trades off between the size of the model (e.g. number of parameters) and the representation size of the encoded residuals. Intuitively, the smaller, less complex, representation is chosen as the preferred model for a given performance criterion. In terms of the robotic systems we consider here, the representation size combining state and model information serves as a measure of system intelligence, and the MRS criterion will select the minimal complexity system for a given task performance. In practice, the MRS criterion has advantages in the attainment of consistent metrics without the introduction of problem specific heuristics or arbitrary weighting factors. The MRS family of methods provides a type of "universal yardstick" for data and models from disparate sources, and therefore has been successfully used in multisensor fusion interpretation problems.

The MRS criterion has been proposed as a general criterion for model inference by Rissanen [19] and by Segen and Sanderson [23]. It is an expression of the ideas on algorithmic information theory pioneered by Solomonoff [24], Kolmogorov [18], and Chaitin [6]. The MRS approach is based on the principle of building the shortest length program that reconstructs observed data. The length of this program or *representation size* depends on both the statistics of the sensors and on the systems "knowledge" of the environment, specified by a set of models and constraints.

More formally, the *representation size* is the length of a program in bits that, when executed on a deterministic Universal Turing Machine (UTM) [7] would reproduce the observed data on the output tape. A model based encoding scheme is used in which the data is thought to be arising from one of the several available models in a model library, Q. The models may differ in structure and number of parameters. The observed data D is encoded by specifying an instantiated model q and the deviations or *residuals* of the data D from the selected model $q \varepsilon Q$. The resulting representation size is

$$L[q,D|Q] = L[q|Q] + L[D|q,Q]$$

$$= L[q|Q] + L[A|q,Q] + L[D|Q,q,Q]$$

where L[q,D] is the total representation size of data D when explained using model q, given a model library Q. L[d|A,q,Q] is the number of bits needed to encode the data deviations or residuals from the model, given a coding algorithm, A. L[A|q,Q] is the number of bits required to specify the coding algorithm itself, given an environment model. L[q|Q] is the number of bits required to encode the environment model (structure and parameters) given a model library, Q.

According to the minimal representation principle, the best explanation of the observed data is the one with the smallest representation size

$$Q_{opt} = \arg\min_{q \varepsilon Q} L[q|Q] + L[A|q,Q] + L[D|A,q,Q].$$

This approach finds the simplest explanation of the data that is most likely, and objectively trades off between model size, algorithm complexity, and observation errors. Rissanen [19] showed that a finite set of random samples from a class of probability distributions would be complexity bounds as defined by Kolmogorov [18] and others [6,24], and the representation size can be used to choose among alternative distribution models. Barron and Cover [4] showed that such a minimal representation size probability distribution is statistically accurate and the rate of convergence is comparable to other methods of parametric and nonparametric estimation. In our previous work [13-17], we have structured the model-based pose estimation problem such that the pose transformation parameters are isolated elements of the statistical model, and may be estimated by the minimal representation criterion.

3. PARTS ENTROPY AND INFORMATION MEASURES FOR ASSEMBLY

Geometric task complexity is directly related to the geometric state space and the precision of state definition or partitioning. In earlier work [20], we have defined the *parts entropy* as a measure of configuration uncertainty in mechanical systems with particular application to assembly analysis and assembly planning. In this formulation, the entropy of a distribution of independent objects, or parts, is given by

$$H_n = H_n (P_1, \ldots, P_n) = - \Sigma P_k \log_2 P_k .$$

where uncertainty in position and orientation is described by the joint probability distribution $P(x,y,z,\alpha,\beta,\chi)$ over the joint ensemble. As an entropy measure [7], H may also be interpreted as the information required to specify the position of the objects in their geometric configuration space.

The part entropy of an object is defined with respect to the mechanically distinguishable positions and orientations, and the resolution, d, in each coordinate degree of freedom. The symmetry of an object therefore strongly affects the resulting orientational entropy and is defined by the set of group operations that leave the object invariant. For example, a sphere has 0 bits of orientational entropy, while a cube with 10 bits of resolution would have 24 bits of entropy.

The part entropy may be used as a basis for the configurational representation size, and is directly related to the set of constraints or other geometric assumptions made on the environment. For example, a flat surface reduces the entropy of parts that sit on it. The entropy of a cube sitting on a table (with 10 bits of resolution) is 28 bits, while a general rectangular solid will be 30.1 bits, and a cylinder may vary from 20 to 30 bits depending on its proportions.

For an assembly task, we consider a set of parts $\{Q_i\}$, l = 1,...,N, such that the part relationships are defined by join probabilities $P[Q_1 ... Q_N]$, and the parts entropy is defined as the joint entropy $H[Q_1 ... Q_N]$. If the parts are positioned independently, for example, prior to assembly, then the probabilities will be independent:

$$P[Q_1 ... Q_N] = P(Q_1)\, P(Q_2) ... P(Q_N),$$

and

$$H[Q_1 ... Q_N] = \Sigma\, H(Q_i).$$

As the assembly task proceeds, individual parts entropies decrease as parts are positioned, and the entropy of the ensemble decreases as part dependence is increased during mating operations. In this sense, an overall goal of the assembly task is to reduce the joint entropy of the ensemble of parts. If we define the entropy of the final rigid assembly to a reference frame with $H_Q = 0$, then the relative entropy of parts and subassemblies may be tracked as a function of time and the entropy flow of the process described in terms of bits per second, that is, information flow. Alternative systems choices and parts designs may be compared in terms of the entropy flow and used to guide decisions on assembly system design. An example described in [20] tracks the parts entropy sequence for sequential assembly for three different electronics assembly strategies. Similar concepts of part probability distributions may be linked to tolerance specifications of assemblies, and have been used to evaluate assemblability based on maximum likelihood methods [21], and used to guide assembly planning tasks [8-10].

4. MULTISENSOR FUSION MANIPULATION EXAMPLE

Figure (2). Five fingered anthropomorphic robot hand manipulating an object. The camera observes motions and minimal representation metrics are used to determine object configuration [16].

The MRS approach has been applied to the problem of multisensor fusion for pose identification of objects using in manipulation by a robot hand. The setting of the task is shown in Figure (2). A five-fingered Anthrobot-3 [2] hand is mounted on a six degree-of-freedom (DOF) articulate PUMA-760 robot arm. The hand is provided with finger tip tactile sensors that sense planar surface contact with the grasped object. The hand is in the field of view of a calibrated camera with edge detection algorithms. A polyhedral object is grasped by the hand and manipulated within the camera view.

In this task scenario, the minimal representation criterion is used to integrate the perception and manipulation steps through the use of consistent information-based criterion for consistency of interpretation of the manipulation with the viewed object pose from the camera. In this task, both the camera information and the tactile sensing data is extremely noisy and uncertain.

The minimal representation formulation of this problem is described in detail in [16]. In this approach, the model-based representation of the hand-eye coordination is described by a set of general constraint equations

$$h(y;z) = 0$$

where Y is a set of model features, and Z is a set of observed data features. In general, such constraints may themselves depend on other model features. Often observed data features may not be related to actual events and identified as unmodeled data features.

The association between the observed data features and the model features is defined by a correspondence w, and this correspondence is a part of the identified model. In addition, a model of the feature extractor, F, for vision and tactile sensing is used to described the process. Application of the MRS approach defines a representation size for each candidate model and set of observations subject to the *data constraint manifold*, DCM, defined by h(y;z). The representation size of the model and encoded residuals is minimized within the measurement subspace locally orthogonal to the DCM.

In general, the search over many candidate models and correspondences is difficult and does not lend itself to linear continuous search techniques. In [16] we use a differential evolutionary algorithm [25] to carry out this search and identify viable interpretations as minimal representation size interpretations of manipulation and sensing states of the system. Figure (3) shows an example of the evolution of the configuration states of the system as the differential evolutionary algorithm proceeds. The system converges to a well-defined and consistent interpretation of the current state (figure (4)).

5. DISCUSSION

The minimal representation size criterion provides a metric for the configurational complexity of robotic tasks and may be used to evaluate alternative algorithms, strategies, and architectures for the accomplishment of specific tasks. The principles of explicit and implict representation are used to define this complexity and the resulting information measures derived may be considered as a measure of configurational intelligence of the system. Specifically, these measures indicate the internal explicit information required to specify the accessible states of the robotic systems using its available perception and actuation capabilities. The resulting approach may be used to evaluate and guide applications tasks such as robotic assembly and multisensor manipulation.

As discussed here, the characterization of tasks is defined with respect to geometric configurations. An important extension of this work is to consider the application of such a formulation to a more general task space involving, for example, force and dynamics of the system requirements.

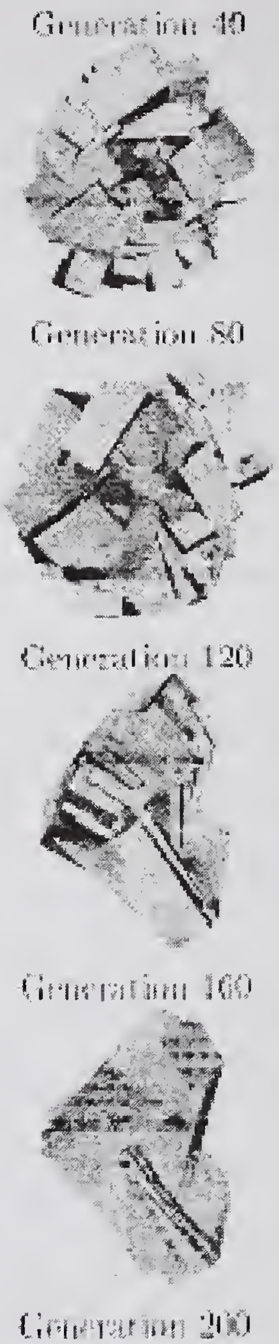


Figure (3). Differential evolution algorithm utilizes representation size metric to search for consistent interpretations of object pose in the hand of manipulator. The minimal representation size pose requires the minimum information to represent.

Figure (4). Final minimal representation pose of the object determined by the differential evolution search.

A second extension of this work is the consideration of intelligent robotic systems with adaptation and learning capabilities. As shown in the multisensor fusion manipulation example, the representation size may be used as a criterion for evolutionary learning of configuration interpretations. In general, this approach might be used to guide learning of algorithmic structure and strategies leading to more sophisticated behaviors.

REFERENCES

[1]. Albus, J.S., H.G. McCain, and R. Lumia, “NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)”, National Bureau of Standards Report SS-GSFC-0027, March 13, 1987.

[2].Ali, M.S., and C. Engler, “System Description document for the Anthrobot-2: A dexterous robot hand,: Tech. Memo 104535, NASA Goddard Space Flight Center, <D, March, 1991.

[3]. Arkin, R. C., *Behavior-Based Robotics (Intelligent Robots and Autonomous Agents),* MIT Press, May, 1998.

[4]. Barron, A. R., and T. M. Cover, “Minimum complexity density estimation”, *IEEE Trans. Inform. Theory,* vol. 37, pp. 1034-1054, July, 1991.

[5]. Brooks, R. A., “A robust layered control system for a mobile robot”, *IEEE Journal of Robotics and Automation*, vol. RA-2, No. 1, March, 1986.

[6]. Chaitin, G. J., “A theory of program size formally identical to information theory,” *J. ACM*, vol. 22, no. 3, pp. 329-340, 1975.

[7]. Cover, T. M., and J. A. Thomas, *Elements of Information Theory*, New York: Wiley, 1991.

[8]. Homem de Mello, L. S. and A. C. Sanderson, "A Correct and Complete Algorithm for the Generation of Mechanical Assembly Sequences", *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 2, pp. 228-240, April 1991.

[9]. Homem de Mello, L. S. and A. C. Sanderson, "Representation of Mechanical Assembly Sequences," *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 2, pp. 211-227, April 1991.

[10]. Homem de Mello, L. S., and A. C. Sanderson, "Two Criteria for the Selection of Assembly Plans: Maximizing the Flexibility of Sequencing the Assembly Tasks and Minimizing the Assembly Time Through Parallel Execution of Assembly Tasks", *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 5, pp. 211-227, October 1991

[11]. Joshi, R., and A. C. Sanderson, "Model-based Multisensor Data Fusion: A Minimal Representation Approach", *1994 IEEE International Conference on Robotics and Automation,* San Diego, CA, May, 1994.

[12]. Joshi, R., and A. C. Sanderson, "Multisensor Fusion and Unknown Statistics using the Minimal Representation Criterion", *Proceedings 1995 IEEE International Conference on Robotics and Automation*, May, 1995.

[13]. Joshi, R., and A. C. Sanderson, ``Multisensor Fusion and Model Selection using a Minimal Re presentation Size Framework", *Proceedings IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, December 8-11, Washington, D. C., USA, Dec. 1996.

[14]. Joshi, R., and A. C. Sanderson, “Experiments in Tactile/Visual Multisensor Fusion”, 1997 IEEE International Conference on Intelligent Robotics, Monterey, CA, July, 1997.

[15]. Joshi, R., and A. C. Sanderson, "Minimal Representation multisensor fusion using differential evolution", *Proc. 1997 Int. Symp. Computational Intelligence in Robotics and Automation,* Monterey, CA., July, 1997

[16]. Joshi, R., and A. C. Sanderson, "Minimal Representation Multisensor Fusion using Differential Evolution", *IEEE Transactions on Systems, Man, and Cybernetics,* January, 1999, Vol. 29, No. 1, pp. 63-76

[17]. Joshi, R., and A. C. Sanderson, *Multisensor Fusion: A Minimal Representation Framework,* World Scientific Publishers, December, 1999.

[18]. Kolmogorov, A. N., "Logical basis of information theory and probability theory," *IEEE Trans. Inform. Theory*, vol. IT-14, no. 5, pp. 662-664. 1968.

[19]. Rissanen, J., "Modeling by shortest data description," *Automatica,* vol. 14, 465-471, 1978.

[20]. Sanderson, A. C., "Parts Entropy Methods for Robotic Assembly System Design", *Proceedings of the IEEE International Conference of Robotics,* Atlanta, GA, March 13-15, 1984.

[21]. Sanderson, A.C., "Assemblability Based on Maximum Likelihood Configuration of Tolerances", *IEEE Transactions on Robotics and Automation*, June, 1999, Vol. 15, No. 3, pp. 568-572.

[22]. Sanderson, A. C., and N. J. Foster, "Attributed Image Matching using a Minimal Representation Criterion", Invited Paper, *Proceedings of the AAAI Spring Symposium,* Palo Alto, California, March 27--29, 1990.

[23]. Segen, J., and A. C. Sanderson, "Model inference and pattern discovery by minimal representation methods," Tech Rept. CMU-RI-TR-82-2, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, July 1981.

[24]. Solomonoff, R.J. " A formal theory of inductive inference," *Inform. Contr.*, vol 7, pp. 1-22, 1964.

[25]. Storn, R., and K. Price, "Minimizing the real functions of the ICEC'96 context by differential evolution," in *Proc. 1996 IEEE Int. Conf. Evolutionary Computation,* May, 1996, pp. 842-844.

[26]. Weiss, L. E., A. C. Sanderson, and C. P. Newman, "Dynamic Sensor Based Control of Robots with Visual Feedback", *IEEE Journal of Robotics and Automation*, Vol. RA3, No. 5, pp. 404-417, October 1987.

Metrics for System Autonomy

Part I : Metrics Definition

Arie ("Arik") Yavnai
Head, Center for Autonomous Systems
RAFAEL, P.O.Box 2250 (39)
Haifa 31021, Israel
Tel: + 972-4-990-8801; Fax: + 972-4-990-8055
e-mail: arikyav@rafael.co.il

ABSTRACT

In many real world applications, system *Autonomy* is the most single significant and meaningful attribute of Intelligent Autonomous Systems - *IAS*. This paper presents performance metrics for *IAS*, which are related to *Autonomy*. Metrics are presented and defined. These metrics are currently being used in on-going research, development and engineering work.

1. INTRODUCTION

From an engineering point of view, performance metrics for *IAS* are needed for establishing and developing the following system level processes: a) a sub-process within the multi-phase system engineering process, e.g., system requirements analysis; b) preliminary and detailed design process; c) Concept-of- Operation development process; d) comparative evaluation of alternative designs.

A fundamental question which is related to *IAS* performance metrics is: Which entity is more meaningful and practical to define and to measure with respect to *IAS* performance – *Autonomy* or *Intelligence*? Our position is that from the user point of view, as well as from the system architect and designer point of view, *Autonomy* is the premier characteristic attribute of an *IAS*. Although *Intelligence* enables *Autonomy*, it is not considered by us as either an appropriate or a practical system design objective or a system performance requirement *per se*.

The concept of *Autonomy* is probably more meaningful, more communicatable, and more precisely measureable, and it is easier to come to a consensus about what *Autonomy* or what an *Autonomous System* is all about, rather than what is *Intelligence* or what is an *Intelligent System*.

2. AUTONOMY

Currently, two distinguished approaches to define system autonomy are used by researchers and groups within the intelligent autonomous systems (including autonomous agents) community. The first approach defines autonomy as an entity which is assigned to the subject system or to the subject agent by a higher level authority, e.g., a supervisor agent. Within the context of this approach, autonomy is defined with respect to the assigned responsibility of a system or an agent. Within this context, autonomy reflects the agent's decision-making capability and authority, and the degree of self control the agent has over its own decisions, see [1]. This approach is more commonly used within the autonomous agents community. The other approach defines system or agent autonomy with respect to its self capability to accomplish its assigned mission goals while operating under uncertain dynamic environment, uncertain dynamic scenario and self faulty situations, and without or with very little human or external agent intervention, [2], [3]. We are using the later approach.

Definition: *Autonomy* is an attribute of a system which characterized its ability to accomplish the system's assigned mission goals without any or with only minimal external intervention, while operating under constraints and under uncertain dynamic environment and scenario conditions.

3. CRITERIA FOR METRICS

In the sequel, some guidelines for metrics selection are proposed.

3.1 *Scope*

The proposed metrics should reflect system autonomy as perceived by an external observer. Therefore, the autonomy should be measured outside the system boundary, i.e., in the interface of the system with external entities. Figure 1, in the sequel, illustrates the context of Autonomy Evaluation, as perceived by an external observer. Four entities are identified within the relevant context, namely: a) a Remote user or supervisor; b) an External Agent; c) Environment & Scenario; d) System Under Evaluation (SUE), which is the Autonomous Intelligent System to be evaluated.

3.2 *Autonomy Relevance*

Meaningful, effective, and measurable metrics for system autonomy should reflect the influence of the following factors as related to system autonomy:

- Level of Abstraction of the commands and the data provided to the autonomous system by the remote user/ supervisor or by an external agent.
- Information bandwidth between a remote user/ supervisor or an external agent, and the system under cvaluation.
- The levels of complexity, dynamics and uncertainty which are attributes to the environment under which the system is operating and executing its mission.
- The levels of complexity, dynamics and uncertainty which are attributes to the system operating scenario while executing its mission.

3.3 *Generality*

Although the meaning of performance metrics is usually domain and application specific, more general entities, such as the principle of *entropy* can be used within the framework of *IAS* performance evaluation. In our work, *entropy* is used as a general measure of entity uncertainty, and is applied to measure various parameters. Using *entropy* as a general tool for representing uncertainty in the domain of control and system engineering was proposed by Saridis [4].

3.4 *Structure Independence*

The metrics for Autonomy should be independent of the internal structure, e.g. : a) number of levels of the hierarchy; b) the decomposition of *IAS* internal processes to resolution scales; c) the computational paradigms, e.g. fuzzy vs. neural networks, and d) other internal specific features. The attempt to establish metrics which takes into account internal specifics of the system will lead to an endless confusing and unpractical effort, and to unstable solution-depended metrics. System Autonomy is a system attribute as perceived by an external observer. In analogy, consider a consumer which want to buy a new car. His decision will not depend on whether the fuel injection control system uses a fuzzy logic based controller or a differential geometry based non-linear controller. However, his decision will probably be based on user-centered parameters such as: fuel consumption (kilometers per liter), number of passengers, safety measures, to name but a few. In such evaluation, the internal specifics are irrelevant. So are the internal specifics when one has to evaluate the performance of an Autonomous Intelligent System.

4. METRICS

In the following section, the metrics used for *IAS* performance evaluation are defined. The nomenclature used is described as follows:

4.1 *Nomenclature*

(1)

Nomcnclature :

ChS - Channel Sensitivity

EnS - Environment Sensitivity

InS - Information Sensitivity

SeS - Seenario Sensitivity

H - Entropy

$H(\Psi)$ *- System Entropy*

$H(\Gamma)$ *- Environment Entropy*

$H(\Lambda)$ *- Scenario Entropy*

C - Channel Capacity of Data Link betwcen Rcmote-Uscr or External Agent to System

Ψ *- System Under Evaluation (SUE)*

Γ *- Environment*

Λ *- Seenario*

I - Externally provided system Information (global and mission related)

Φ *- Remote Uscr*

Ω *- Problcm Contcxt*

n - Time step index

4.2 *Entropy*

We are using *entropy* as a measure of uncertainty of system state, environment state, or scenario state. The uncertainty associated with predicting the next entity state, given the current entity state, is a measure of the entity irregularity or 'disorder'. The less is the entity regularity, the greater is the next state prediction uncertainty and the greater is the associated entropy. Thus, entropy can be used as a measure of environment uncertainty as well as a measure of scenario uncertainty. Entropy can also be used as a measure of system uncertainty, which is directly related to system performance. It can represents the uncertainty in selecting the appropriate control from the set of all admissible controls [4]. Entropy can also be used for representing performance, e.g., system tracking error along a planned trajectory in the system state space.

We define *entropy* as follows:

(2)

Entropy Definition

$$P(X,n,l) = Prob\ \{X(n+1)=X_l \mid X(n)\}\ ;$$

$$X_l \in \{X\}$$

$$H\ (X,n) = - \sum_{l} P(X,n,l) \bullet ln\ P(X,n,l)$$

X - Entity State -

(e.g., best control action; Environment State; Seenario State)

H - Entropy

$H(\Psi)$ *- System Entropy*

$H(\Gamma)$ *- Environment Entropy*

$H(\Lambda)$ *- Seenario Entropy*

4.3 Channel Sensitivity

Channel Sensitivity- ChS, is defined as the differential change of the system entropy which results after a differential change in the channel capacity of the information data link between a remote-user and the System Under Evaluation - *SUE*, or between an external agent and the *SUE*, has occurred.

(3)

Channel Sensitivity :

$$(ChS)_n = \frac{\Delta H(\Psi,n))/H(\Psi,n)}{\Delta C(n)/C(n)}$$

$|| \ \Psi=\Psi_\mu;\ C=C_\phi;\ \Gamma=\Gamma_\alpha;\ \Lambda=\Lambda_\delta;\ \Phi=\Phi_\beta$

$$\overline{ChS} = \frac{1}{n} \sum_{k=1}^{n} (ChS)_n$$

$\Psi_\mu \in \{\Psi\};\ X_\phi \in \{X\};\ \ \Gamma_\alpha \in \{\Gamma\};$

$\Lambda_\delta \in \{\Lambda\};\ \Phi_\beta \in \{\Phi\};\ \Omega = (\Gamma,\ \Lambda,\ \Phi, I)$

Definitions:

If $\overline{ChS} \prec 0 \Rightarrow$ *SUE is Non-Autonomous w.r.t. C,*
under context Ω

If $\overline{ChS} \equiv 0 \Rightarrow$ *SUE is Autonomous w.r.t. C ,*
under context Ω

If $\overline{ChS} \succ 0 \Rightarrow$ *SUE is Non-Supervisable w.r.t. C,*
under context Ω

4.4 Environment Sensitivity

Environment Sensitivity- EnS, is defined as the differential change of the system entropy which results after a differential change in the environment entropy, or uncertainty, has occurred.

(4)

Environment Sensitivity :

$$(EnS)_n = \frac{\Delta H(\Psi,n)/H(\Psi,n)}{\Delta H(\Gamma,n)/H(\Gamma,n)}$$

$|| \ \Psi=\Psi_m;\ C=C_f; \Gamma=\Gamma_a;\ \Lambda=\Lambda_d;\ \Phi=\Phi_b$

$$\overline{EnS} = \frac{1}{n} \sum_{k=1}^{n} (EnS)_n$$

$\Psi_m \in \{\Psi\};\ \ C_f \in \{C\};\ \Gamma_a \ \in \{\Gamma\};$

$\Lambda_d \in \{\Lambda\};\ \Phi_b \in \{\Phi\};\ \Omega = (\Lambda,\ \Phi, C, I)$

Definitions:

If $\overline{EnS} \succ 1 \Rightarrow$ *SUE is Non-Autonomous w.r.t.* Γ,
under context Ω

If $0 \prec \overline{EnS} \leq 1 \Rightarrow$ *SUE is Partly Autonomous w.r.t.* Γ,
under context Ω

If $\overline{EnS} \equiv 0 \Rightarrow$ *SUE is Completely Autonomous w.r.t.* Γ,
under context Ω

4.5 Scenario Sensitivity

Scenario Sensitivity- ScS, is defined as the differential change of the system entropy which results after a differential change in the scenario entropy, or uncertainty, has occurred.

(5)

Seenario Sensitivity :

$$(SeS)_n = \frac{\Delta H(\Psi,n)/H(\Psi,n)}{\Delta H(\Lambda,n)/H(\Lambda,n)}$$

$|| \; \Psi=\Psi_m; \; C=C_f; \Gamma=\Gamma_a; \; \Lambda=\Lambda_d; \; \Phi=\Phi_b$

$$\overline{ScS} = \frac{1}{n}\sum_{k=1}^{n}(ScS)_n$$

$\Psi_m \in \{\Psi\}; \;\; C_f \in \{C\}; \; \Gamma_a \in \{\Gamma\};$

$\Lambda_d \in \{\Lambda\}; \; \Phi_b \in \{\Phi\}; \; \Omega= (\Gamma, \Phi, C, I)$

Dcfinitions:

If $\overline{ScS} \succ 1 \Rightarrow$ *SUE is Non-Autonomous w.r.t.* Λ,
undcr contcxt Ω

If $0 \prec \overline{ScS} \leq 1 \Rightarrow$ *SUE is Partly Autonomous w.r.t.* Λ,
undcr contcxt Ω

If $\overline{ScS} \equiv 0 \Rightarrow$ *SUE is Complctcly Autonomous w.r.t.* Λ,
undcr contcxt Ω

4.6 *Information Sensitivity*

Information Sensitivity- InS, is defined as the differential change of the system entropy which results after a differential change in the system global and mission related externally provided information, has occurred. The information includes the Mission Plan and the related Data Bases which provided to the autonomous system by the remote user/ supervisor or by an external agent, prior to mission execution, or while the mission is executed.

(6)

Information Sensitivity :

$$(InS)_n = \frac{\Delta H(\Psi,n))/H(\Psi,n)}{\Delta I/I}$$

$|| \; \Psi=\Psi_m; \; C=C_f; \;\; \Gamma=\Gamma_a; \;\; \Lambda=\Lambda_d; \; \Phi=\Phi_b$

$$\overline{InS} = \frac{1}{n} \sum_{k=1}^{n} (InS)_n$$

$\Psi_\mu \in \{\Psi\}; \;\; X_\phi \in \{X\}; \;\; \Gamma_\alpha \in \{\Gamma\};$

$\Lambda_\delta \in \{\Lambda\}; \;\; \Phi_\beta \in \{\Phi\}; \;\; \Omega= (\Gamma, \Lambda, \Phi, C)$

Dcfinitions:

If $\overline{InS} \succ 1 \Rightarrow$ *SUE is Non-Autonomous*
w.r.t. I, under context Ω

If $0 \prec \overline{InS} \leq 1 \Rightarrow$ *SUE is Partly Autonomous*
w.r.t. I, under context Ω

If $\overline{InS} \equiv 0 \Rightarrow$ *SUE is Completely Autonomous*
w.r.t. I. under eontext Ω

4.7 Adaptation Rate Sensitivity

Adaptation Rate Sensitivity - ARS, is defined as the differential change of the system entropy rate which results after a differential change in the entropy of the subject entity, e.g., environment or scenario, or uncertainty, has occurred. Similarly, Adaptation Rate Sensitivity can be defined in relation with differential changes of channel capacity. or information.

(7)

Adaptation Rate Sensitivity:

$$(ARS)_n = \frac{\Delta(\partial H(\Psi,n)/\partial n)/(\partial H(\Psi,n)/\partial n)}{\Delta H(X,n)/H(X,n)}$$

$$\| \ \Psi=\Psi_m;\ C=C_f;\Gamma=\Gamma_a;\ \Lambda=\Lambda_d;\ \Phi=\Phi_b$$

$$\overline{ARS} = \frac{1}{n}\sum_{k=1}^{n}(AdR)_n$$

$$\Psi_m \in \{\Psi\};\ \ C_f \in \{C\};\ \Gamma_a \in \{\Gamma\};$$

$$\Lambda_d \in \{\Lambda\};\ \Phi_b \in \{\Phi\};\ \Omega = (\Gamma,\ \Phi, C, I)$$

Dcfinitions:

If $\overline{ARS} \succ 1 \Rightarrow$ *SUE is Non-Autonomous w.r.t. X,*
undcr contcxt Ω

If $0 \prec \overline{ARS} \leq 1 \Rightarrow$ *SUE is Partly Autonomous w.r.t. X,*
undcr contcxt Ω

If $ARS \equiv 0 \Rightarrow$ *SUE is Complctcly Autonomous w.r.t. X,*
undcr contcxt Ω

5. SUMMARY

Metrics for system autonomy has been defined and presented. Following the metrics, a specific measure for a certain application can be derived directly. Associated with each definition, the broad classification of the SUE was defined.

REFERENCES

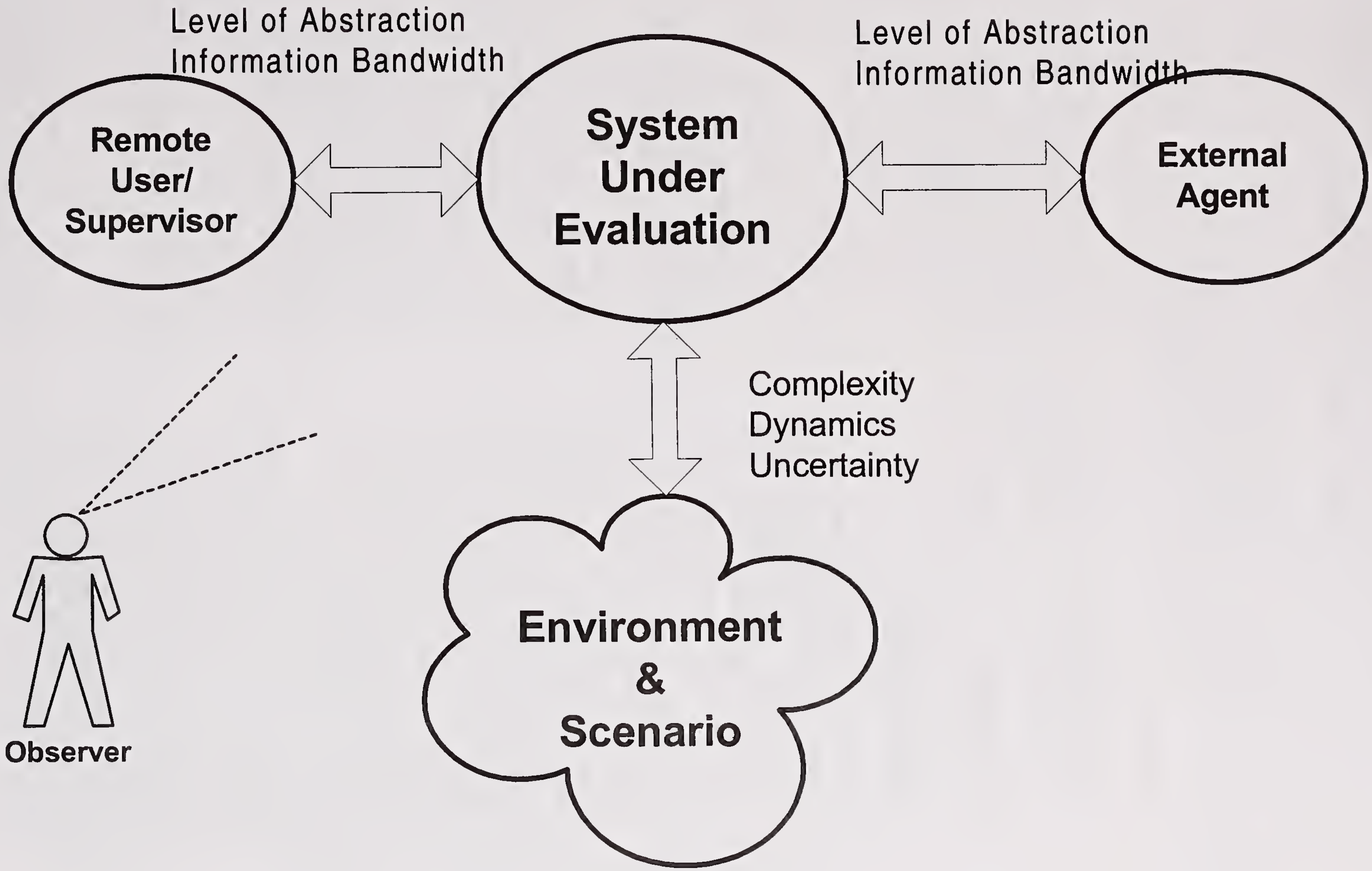
[1] Barber, K.S., (1999), "Agent Autonomy: Specification, Measurement, and Dynamic Adjustment", 1999, pp. 1-8.

[2] Yavnai, A., (1989), "Criteria of System Autonomability", Proc. Intl' 2nd Conference on Intelligence Autonomous Systems, Amsterdam, Dec. 1989, pp. 448-458.

[3] Yavnai, A., (1991), "Entropy-based Criteria for Iintelligent Autonomous Systems", Proc. IEEE Intl' Symp. Intelligent Control 1991, Arlington VA., August 1991, pp. 55-60

[4] Saridis G. N. (1995), "Stochastic Processes, Estimation and Control - The Entropy Approach", Wiley-Interscience, New-York, NY, 1995.

CONTEXT OF AUTONOMY EVALUATION



In Defense of the Additive Form for Evaluating the Multidimensional Vector

Dr. Leonid M. Polyakov
Globe Institute of Technology
Leonid@globeinstitute.org

ABSTRACT

The topic of this discussion is an artificial (not a natural human) intelligence measurement. It would be better to call it an evaluation rather than a measurement. The Additive Evaluation Method is the **only real** method to make a evaluation of the vector value.

KEYWORD: *intelligence, measurement, expert. additive.*

ADDITIVE FORM.

Artificial Intelligence, like a human one, is a composition of the different additive abilities such as reasoning, learning, decision-making, object recognition, and so on. The multifunctional nature of intelligence can be represented as a **vector** .

The intelligence measurement is not the same as a multiobjective optimization of the intelligence systems. There are many different methods of optimization (Preference Structures, Compromise Approach, Lexicographic Ordering Approach, Genetic Approach, Pareto approach, etc.) [4,5, and other]. All of these methods work with each function of the intelligence separately and determine preferences and a system's rank, but not an intelligence value. The additive function is presented in the most of the research works [2,3,6,7,9-14, and other].

The measurement is a process of **assigning numbers** to the objects or events in accordance with certain rules of the system. The number assignment is possible just on the scalar scale. There are three types of axioms related to a measurement process: identity axioms, rank axioms, additivity axioms. These axioms determine four scale levels: scale of names, range scale, interval scale and ratio scale. The analyses of these scales are done in [2]. Only additivity axioms can be applied to the real measurement. These axioms can be applied just to the **scalar** scale, as it was mentioned above. A vector doesn't meet these conditions. Just, the weighted-sum approach and utility functions can be used in this case [3,7] as the method of multivariable scales aggregation and converts vector into a sufficient scalar.

The last question is how to determine the value of weight. The most known and usable method is an expert method, but there are several **analytical methods** to find out the value of this function [2,6].

Opponents of these methods of the aggregation function complain against the application of a human expertise as a source of information. They dispute an expert ability to produce objective information. Yes, a collective expertise has an element of subjectivism but today we don't have a better way to measure a vector's values to make a comparison of two or more vectors' values. Is this, a wonderful fact in that we use an expert's intellectual ability in the intelligence measurement? Certainly not, because the intelligence can be measured by the scale of the intelligence. Only a human being has the best sense of the value of the intelligence functions. Each separate intelligence function can be measured by appropriate methods but, as an integrated value, intelligence has to be presented as a scalar.

There are many different methods to measure each separate intellectual ability. For example, the value of the ability to learn can be presented as a ratio of an increment of intelligence to an increment of information. The number of iterations, or the number of rules and trials (trial and error method), or the entropy method, etc can determine the value of information. So, the learning ability is:

$$L = d(I)/d(If). \qquad (1)$$

The amount of new information available to the different systems can change the intelligence value of these systems.

A values of a separate intellectual abilities (variables) don't give any ideas about artificial intelligence integrated value. Aggregation of the separate variables can be done on the base of the utility theory. The utility of intelligent alternative can be presented as [2]:

$$U_A = \sum_{i=1}^{n} U_i \qquad (2)$$

where *Ui* is an utility of *i-th* basic variable,
n is a number of variables.

From (2) [2], we can get the quality index of j-th alternative (domain specific by design) in nondimensional units

$$Q_j = \sum_{i=1}^{n} W_i (F_i)*(Fi/Fi \max)\ (3)$$

Where *Wi (Fi)* is a weight function of *i-th* variable *(Fi)*.

A set of variables has to be named for each problem separately.
Usually one of the variables is an investment value of the j-th alternative (C_j). In this case, equation (4) can be rewritten as:

$$Q_j*(C \max/ Wc) = \sum_{i=1}^{n-1} [Wi (Fi)/Wc] * C_{max}*(Fi/Fimax)- C_j. \ (4)$$

This equation presents the evaluation of j-th alternative measured in cost units (dollars). Now we can use money as a real universal scale of the measurement. Some opponents can say, "it is immoral". A measurement is not a moral category! $Ç_j$ can be added to the left and the right parts of the equation (4). In this case we can get the value of $Q_j*(C \max/ Wc)$ presented in dollar units. This value includes only intelligence variables and can be called the intelligence value of the j-th
alternative

$$I_j = \sum_{i=1}^{n-1} [Wi (Fi)/Wc] * C_{max}*(Fi/Fimax). \qquad (5)$$

Where *Wc* is a weight function of variable *Cmax*.

This is the direct way to calculate **profit** (political factors are included). It is **one more reason** to use the Utility Method and scalar scale. **No other method permits us to get an intelligence evaluation in dollar units.** Each time in the shopping center, when we are buying something we use ours preferences and convert a vector value into a scalar value presented in the dollar units.

The intelligence measurement is not a new problem. The famous IQ and WAIS-3 [8] tests are the possible ways to make an evaluation of the human intelligence. These tests present an aggregated value of the multifunctional intelligence and convert a vector value into a scalar value.

The opponents to these testes pointed out to the possible social problems bounded to these methodic. In case of artificial intelligence measurment this problem does not make sense.

Conclusion.

The Additive Evaluation Method is the **only real** method to make a evaluation of the vector value. It can't be write off from the tools of intelligence value evaluation. Artificial intelligence of the system should be measured and presented as scalar.

This method is the **only one,** which can gives financial evaluation of artificial intelligence application.

Contemporary artificial intelligence systems are design as a domain-oriented systems. **Only the expert** can determine the importance of each intellectual function with regard to the certain domains.

References:

1. Pareto M. D'economic Politque. Paris 1927, 695p.

2. Polyakov L.M., Kheruntsev P., Shklovsky B., Elements of the Automated design of the electrical automated equipments of Machine tools. Publ. "Machinostrojenie", Moscow, 1974, 157p.

3. John Von Newmann. O. Morgenstern, Economic Behavior and Theory of Games, 1944, 650p.

4. Dhar, V. Stein, R. Intelligent Decision Support Methods Prentice Hall, 1997, 244p.

4. Mitsuo Gen, Runwei Cheng. Genetic Algorithms Engineering & Optimization. A Wiley-Interscience Publication, 2000, 490p.

5. Sigford Y., Pazvin R., Project PATTERN: a methodology for determining relevance in complex decision making. IEEE Transaction. Eng. Manag. V.EM-12, No.1, 1965, 210p.

6. Fishburn P. C., Additive Utilities with Incomplete Product Set: Applications to priorities and Assignments- Operations Research, V. 15, 1967, No.3. p.537-542.

8. Charles G. Morris, Albert A. Maisto, Psychology. Prentice Hall, 1999, 682p.

9. Marino Y. P. Technological forecasting for decision making, American Elsevior Company Inc. N.Y., 1972.

10. Fishburn P. C. A Study of Independence in Multivariate Utility Theory. Econometric, 37, No.1, 1969, p.7-121

11. Fishburn P.C. Independence in Utility Theory with Whole Product Sets. Operations Research, V.13, 1965, p.28-45.

12. Schziver A. Forecast Air Review, V. 16, No.3, 1965, p.12-23.

13. Larichev O. The Art and Science cf Decision Making, "Nauka", Moscow, 1979, 196p.

14. Hall A. Experience of Methodology for Large Engineering Systems, "Soviet Radio," M, 1975, 120p.

A Method For Evaluating the "IQ" Of Intelligent Systems

Robert Finkelstein

University Of Management & Technology
1925 N. Lynn Street, 3rd Floor
Arlington, Virginia 22209
(703)-516-0035 Ext. 24
And
Robotic Technology Incorporated
11424 Palatine Drive
Potomac, Maryland 20854-1451
(301)-983-4194
RobertFinkelstein@compuserve.com

ABSTRACT

This paper describes a **pragmatic** process measuring the "IQ" of individual intelligent robots and groups of intelligent robots. We offer definitions for the characterizations "intelligence" and "IQ." We define **metrics and submetrics** for individual robots and the collective, using the Analytic Hierarchy Process (AHP) to calculate weights for the metrics and submetrics. They can then be used to evaluate alternative technologies and systems for achieving individual and collective intelligent behavior in robots.

The defined metrics and submetrics for **individual** robots include:

- ❑ **Intelligence** (decomposed into the ability to make Correct Decisions and Right Decisions, and to Learn)
- ❑ **Effectiveness** (decomposed into the ability to achieve Objectives, Goals, and Priorities)
- ❑ **Efficiency** (decomposed into Accuracy, Precision, Time efficiency, Energy Efficiency, and Side Effects)

The defined metrics and submetrics for **groups of robots** include:

- ❑ **Command And Control** (decomposed into Leadership, Followership, and Efficiency)
- ❑ **Communications** (decomposed into Message Initiation, Transmission, Understanding, and Efficiency)
- ❑ **Effectiveness** (decomposed into the ability to achieve Objectives, Goals, and Priorities)

The values of the metrics are determined by experimenting with designed robotic systems, in the context of a scenario, in a simulation or field experiment. The weighted metrics can be combined to obtained an "IQ" score for individuals or the collective.

Keywords: *Intelligent Robots, Metrics, IQ, Intelligent Systems, AHP, Robot Groups*

1. Introduction

A little learning is a dangerous thing;
Drink deep, or taste not the Pierian spring;
There shallow draughts intoxicate the brain
And drinking largely sobers us again.
----- Alexander Pope

The brain is the most overrated organ.
----- Woody Allen

There are no satisfactory definitions of *human* intelligence, so it is not surprising that are no generally accepted definitions of *machine* intelligence either. One implicit definition, "the ability to cope with the unexpected, and the ability to bring about the unexpected," is from a comment in the *Economist* about the major attribute for a good U.S. president. Another suitable definition of intelligence, is **"the ability to make an appropriate choice or decision."** The intelligence need not be at the human level. The ability to make an *appropriate* choice is common to all long-time survivors, including roaches and rats. *Appropriate*, for organisms, usually means enhancing the ability to reproduce, the primal goal in life. A chicken is an egg's way of making another egg. *Appropriate* intelligence for a robot might mean the ability to accomplish its mission under a variety of conditions.

It has been difficult to measure human intelligence in a satisfactory way since the first "IQ" (Intelligence Quotient) tests were developed at the start of the 20th century. The tests, and their interpretations, remain controversial. The measurement of machine intelligence, however, is a somewhat easier task primarily because (1) the functional domains of interest are more narrowly defined than for people and (2) the underlying mechanisms of machine intelligence are more accessible to experimentation than the means of human intelligence.

When intelligent machines are designed for a limited set of functions - such as performing search and rescue of people trapped in collapsed buildings - they can be tested within that small sphere of endeavor regardless of the intended level of their intelligence (e.g., whether they are as intelligent as insects or humans). In this sense, the robot's IQ test is analogous to a person's aptitude test for a job or profession. More importantly: for measuring the "IQ" of an intelligent machine, the tester has access to the underlying intelligent control system. This allows the intelligent control system to be connected to an avatar of the machine in a simulated environment. The tester can know the ground truth - have a "god's-eye view" - of everything in the environment, including every external manifestation of behavior by the robot avatar as well as the control system's every internal state (including learned and adaptive behavior). The intelligent control system need not know - or care - that it is controlling an avatar in a simulation and not a physical system in the real world. Of course, the validity of the simulation is only as good as the ability of the underlying model to replicate the real-world environment of interest.

Intelligence can be decomposed into the ability to make a *correct* decision (the optimum decision given complete knowledge), a *right* decision (the optimum decision given limited knowledge), and *learning* (the ability to adapt to the environment, without necessarily making a decision which leads *immediately* to altered, observable external behavior). While *intelligence* is an important metric (measure of merit) for an autonomous intelligent robot, there are two other key measures (as per Peter Drucker): *efficiency* (a measure of how well the autonomous robot does things right) and *effectiveness* (a measure of how well the robot does the right thing). These metrics take into account other system variables and characteristics, including: energy expenditure, mobility, reliability, stealthiness, etc. An intelligent robot with a failed engine or damaged servo motors cannot move to accomplish its mission no matter how well it has planned its path. Some researchers are redefining human IQ to include a variety of human talents, including physical skills. Indeed, some anthropologists believe that human intelligence was quite fragmented and narrowly focused task by task (as in *Homo neanderthalensis*) until recently, when intelligence became synthesized in *Homo sapiens sapiens*. Likewise, the "IQ" of an intelligent robot might include the **combined metrics of cognitive and physical abilities**: Intelligence, Effectiveness, and Efficiency. However they are labeled or amalgamated, these metrics can be quantified and used to test the performance - the "IQ" - of any autonomous intelligent system.

1.1 Sundry Definitions Of Intelligence

"Civilization advances by extending the number of important operations which we can perform without thinking."
— Alfred North Whitehead

For organisms, intelligence is a pragmatic mechanism of survival; and all measures of intelligence (whether for organism, man or machine; whether genetically encoded, pre-programmed, or learned) involve an **ability to make appropriate selections** [1, 2], **choices, or decisions**. Human intelligence involves "the degree to which an individual can successfully respond to new situations or problems. It is based on the individual's knowledge level and the ability to appropriately manipulate and reformulate that knowledge (and incoming data) as required by the situation or problem," [3]. Intelligence can be identified by an ability to cope with the unexpected and an ability to bring about the unexpected, abilities against which to judge presidents, among other notables, over history [4]. The subjective word "appropriate," in relating intelligence to "appropriate" choice, implies that a system can be intelligent only in relation to a defined goal or environment.

Intelligence requires an ability to use information (where information, according to Claude Shannon, is that which reduces uncertainty) [2], and using information includes an ability to detect new, non-chance associations [5]. Chen defines intelligence (individual or organizational) "as the attainment of relevant goals in specified contexts using appropriate means and resulting in positive outcomes," [6], which is the same as saying, as above, "intelligence is the ability to make an appropriate choice."

A behaviorist would say, in the spirit of the Turing Test, that if humans, machines, or organizations (collectives) **behave** intelligently, then they are; if they manifest consciousness, then they are conscious. Two parts of intelligence are: (1) epistemological, in which the world is so represented that solutions to problems follow from the facts expressed in the representation; and (2) heuristic, in which there is the mechanisms that solves the problem and selects actions on the basis of information (most work in artificial intelligence is devoted to the heuristic part) [7]. Entities can place different emphasis on these two kinds of mechanisms of intelligence, depending on the context.

In one view [8], attributes of systems with higher intelligence include:

❑ mental attitude (beliefs, intentions, desires);
❑ learning (ability to acquire new knowledge);
❑ problem solving;
❑ understanding (implications of knowledge);
❑ planning and predicting consequences of actions, comparing

alternative actions;
- ❑ knowing limits of knowledge and abilities;
- ❑ drawing distinctions between similar situations;
- ❑ synthesizing new concepts and ideas, acquiring and employing analogies, originality;
- ❑ generalizing;
- ❑ perceiving and modeling the external world;
- ❑ understanding and using language and symbolic tools.

Some hold that the last attribute - language - is the prime determinant of higher intelligence; that every representation of knowledge is an interpretation, not decision-making or expertise [9]. Symbolic manipulation [10] - communication - creates second order reality; an advanced, intelligent system must be able to perceive second order reality (the meaning and values attributed to first order reality) as well as first order reality (the reality accessible to perceptual consensus, physical reality) [11].

A machine with higher intelligence should be able to: adapt (changing itself or the environment) for survival; reason about its own organization, reasoning ability, and external domains; plan internal activities (database searches, decision-making) and external activities (sending messages, physical actions); select among decision-making processes; make decisions using values associated with possible actions; reason about its reasons for taking actions; value itself to avoid changing itself in a harmful way. Ideally, the system should be self-conscious as well as self-adaptive [12].

The higher intelligent system should possess meta-knowledge, i.e., it should have knowledge about what it knows without having to search exhaustively. For example, the system should know whether it has knowledge about grapefruit if asked the size of grapefruit [13]. Knowledge includes representations of facts, generalizations, and concepts, organized for future use [5]. Knowledge of general truths does not require a special metaphysically distinct ingredient in humans [14] - machines can be designed to know such truths. "Knowledge is more than a static encoding of facts; it also includes the ability to use those facts in interacting with the world ... knowledge of something is the ability to form a mental model that accurately represents the thing as well as the actions that can be performed by it and on it. Then by testing actions on the model, a person (or robot) can predict what is likely to happen in the real world," [15].

"The use or handling of knowledge" is cognition [16], "an intellectual process by which knowledge is gained about perceptions or ideas," [17]. An intelligent system can be designed to learn ("any deliberate or directed change in the knowledge structure of a system that allows it to perform better on later repetitions" of a task [18]). But it would be difficult to give it common sense, which involves a larger variety of different types of knowledge than expertise (a large amount of knowledge of relatively few varieties) [19]. A robot is *behaving* consciously if it [20]:

- ❑ receives information about its environment;
- ❑ recalls and compares past experiences;
- ❑ evaluates the quality of its experiences;
- ❑ makes conceptual models of its environment;
- ❑ projects consequences of alternative future actions;
- ❑ chooses and implements actions which further its goals.

By exhibiting purpose and intention, a machine would behave *as if* it had free will and the ability to choose [21].

1.2 Group Intelligence

Organizations or collectives can become intelligent through the emergent behavior of its organisms and machines. "Emergent behavior involves the repetitive application of seemingly simple rules that lead to complex overall behavior," [22]. The emergent behavior can be that of an ant and ant colony, a person and an organization, or a robotic vehicle and a combat platoon. Collective intelligence in insect societies, especially for certain of the ants, bees, and termites, is reasonably understood. "Higher forms of intelligence arise from the synchronized interaction of simpler units of intelligence," [23]. This is true as well of "higher" forms of life, such as dolphins, wolves, apes, and humans. Social intelligence allows an individual organism to "analyze and respond correctly (intelligently) to possible behavioral responses of other members of the group," [24]. Collective intelligence, an advanced form of intelligence, "involves group intelligence in which individuals submerge their individual identity" [24] to the group's responses to the threats and opportunities in the environment. Communication among individuals is essential for collective intelligence, whether by pheromone, vision, sound, touch, or email. Information technology is now affecting the collective intelligence and evolution of the human species, possible leading to the emergence of a global intelligence, a system of individual and collective humans and machines [25]. But, as always, the essence of intelligent behavior is control - at least *self-control.*

There have been a number of programs attempting to develop cooperative mobile robots, and over 200 papers have been published concerning mobile cooperative robots [26]. Cao, Fukunaga, and Kahng [27], on which much of the following discussion of cooperative behavior is based, define **collective behavior** generically as "any behavior of agents in a system having more than one agent," while **cooperative behavior** is defined as "a subclass of collective behavior that is characterized by cooperation." Cooperation should lead to the enhanced performance of the collective over that of the simple aggregation of individuals (i.e., the whole should be greater than the sum of its parts). Cao et al. cite the following definitions of cooperative behavior (from various sources):

- ❑ To associate with another or others for mutual, often economic, benefit.

❑ Joint collaborative behavior that is directed toward some goal in which there is a common interest or reward.

❑ A form of interaction, usually based on communication.

❑ Joining together for doing something that creates a progressive result, such as increased performance or saving time.

❑ Given some task specified by a designer, a multiple robot system displays cooperative behavior if, due to some underlying mechanism (i.e., the mechanism of cooperation) there is an increase in the total utility of the system.

As posed by Cao et al., the fundamental issue is: *given a group of robots, an environment, and a task, how should cooperative behavior arise?*

The architecture of a computing system is that part which remains unchanged unless an external agent changes it. The group architecture of a cooperative system is the infrastructure on which collective behaviors are implemented and determines the abilities and constraints of the system. The group architecture for cooperative robots includes such considerations as: robot heterogeneity and homogeneity, the ability of each robot to recognize and model other robots, and communications. Also, the architecture must be able to avoid conflicts among robots for resources, such as paths through the environment, goal objects in the environment, and communications bandwidth.

In creating a group architecture, there are a number of alternative design decisions. The architecture may be centralized or decentralized. Centralized architectures are characterized by a single control agent. Decentralized architectures, which are prevalent, may be either distributed or hierarchical. In the former, all agents are equal with respect to control, while the latter are locally centralized. Decentralized architectures may lead to emergent properties of systems, such as intelligence or self-organization. Their inherent advantages over centralized architectures include fault tolerance, natural exploitation of parallel processes, reliability, and scalability. Most robot architectures hybrid, where, for example, a central planner exerts high-level control over mostly autonomous agents. A group of robots is homogeneous if the capabilities of the individual robots are identical; otherwise they are heterogeneous (forming a more complex system).

Cooperation among robots can arise from eusocial behavior (as opposed to explicit cooperative behavior) which results from the behavior of individuals and not necessarily an a priori effort at cooperation (e.g., ants and bees are eusocial). There are many sorts of self-organizing systems (in which there has been much research), but especially with respect to biological systems, whether individual organisms (in which the individuals parts are self-organizing) or social groups (human or otherwise). The aggregation of relatively limited individuals leads to the collective's more capable intelligence (this is true of human society as well). Individual robots that are selfish and utility-driven, but must cooperate in order to survive, will display emergent cooperative behavior. Explicit cooperation, as among humans, can be driven by a desire to maximize individual utility, so there are economic and game-theoretic approaches to examining cooperation.

It is difficult for human designers to account for the multiplicity of control variables and contingencies to achieve cooperative behavior in robots. It is easier to design the robots so that they learn to cooperated and adapt to their environment. A number of techniques are being developed for this approach, including the use of neural networks and genetic algorithms.

The robotic group may employ various types of communications processes for inter-agent interaction, including, in one taxonomy: interaction by means of the environment; interaction by sensing; interaction by explicit communications. The simplest, most limited type of interaction occurs when the environment itself is the communications medium, providing the equivalent of a shared memory among a group of robots. There is no explicit communication or interaction among the individuals.

Another form of group communications occurs when individuals sense and perceive one another without engaging in explicit communications. Using a suitable sensor (e.g., vision, acoustic, chemical, touch), the individuals must be able to distinguish members of the group from other entities in the environment. Resulting collective behavior includes flocking and pattern formation relative to neighboring individuals.

Higher-order tactical group behavior generally requires explicit communication among individuals, which can be directed (to known recipients) or broadcast (to unknown recipients). Architectures that enable this type of communication resemble communications networks, and communications protocols are necessary for inter-robot communications. The message carrier can consist of various portions of the electromagnetic spectrum (e.g., radio frequency, microwave, optical, infrared) or other transmission mechanisms (e.g., acoustic, chemical).

In order to function relative to others in a group, or with respect to predators (threats) and prey (targets), individual organisms (or robots) must be able to model the intentions, beliefs, actions, capabilities, and states of those others. The ability of individuals to model others in a group reduces the need for communications; it encompasses implicit communications via the environment and perception and includes representations of other individuals which can be used to make inferences about the actions of those individuals.

There are many prospective means of achieving cooperative behavior among robots. The most direct is to explicitly program the desired behavior. This is difficult and tedious in that the programmer must a priori account for all possible contingencies. Other methods are more promising, including biological (e..g, social insects) behavioral approaches, task decomposition and allocation approaches, game-theoretic approaches, machine learning approaches, and approaches based on cooperation as an emergent property of complex group dynamics. Geometric approaches include multi-agent path planning, moving to formation, and pattern generation.

For most military applications, explicit leader-follower relationships are important, especially where robotic forces will be integrated with conventional forces. These roles and abilities may exist in all of the robots, where leaders are anointed - or emerge - based on circumstance (as is often the case for people). Or leaders may be specially trained as such.

For example, group behavior to achieve coordinated movement in the world, such as path planning, can be centralized (with a leader or universal path planner making decisions) or distributed (with individual agents planning and adjusting their paths). They may be hybrids, combining on-line, off-line, centralized, and decentralized elements. Planning systems may take into account all robots, or plan the path of each one independently. Factors include dynamically-varying global and individual priorities, environmental constraints and obstacles, and the allocation of space-time resources. Conflicts may be resolved by a central manager or negotiated among individuals.

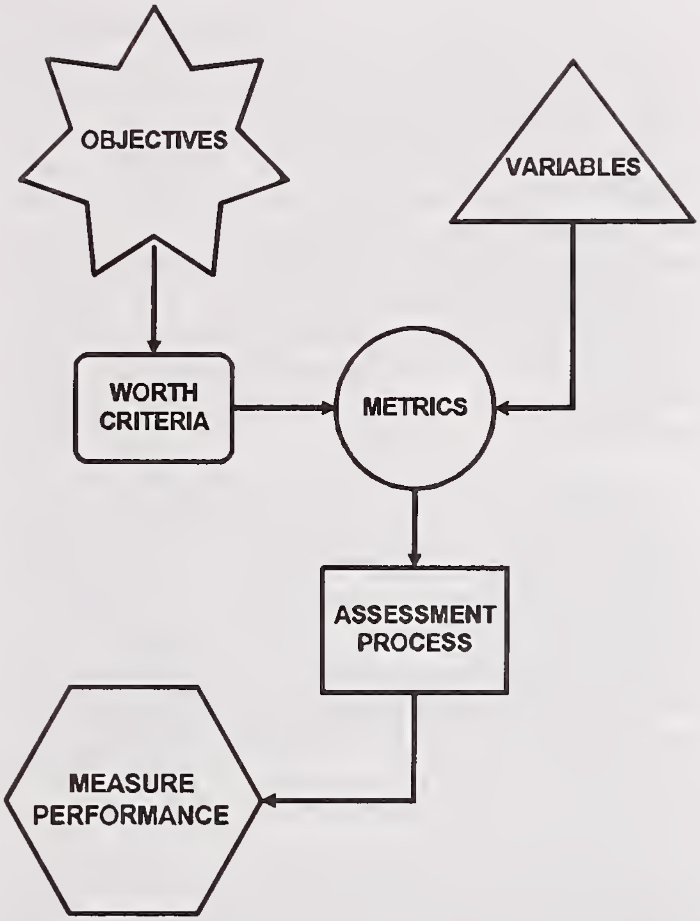
2.0 Evaluation Process

In order to evaluate the performance of an intelligent robot (or group of robots) we can employ a *pragmatic*, behaviorally-based, teleological, and functional approach to measuring its "IQ" as follows:

- ❑ Define the purpose or mission or objectives of the robot (or group of robots)
- ❑ Derive the worth criteria by which the robot's performance may be assessed
- ❑ Organize and integrate the worth criteria into a consistent assessment structure
- ❑ The assessment structure employs suitable variables (endogenous, status, and exogenous), metrics and submetrics, a means of weighting or ordering the metrics and submetrics, and a means of evaluating performance against the ordered metrics and submetrics.
- ❑ Measure the performance of the robot or group of robots, in the context of the desired scenario and environment, in a simulation or field exercise, calculating "IQ" from the evaluated, weighted metrics and submetrics.

The evaluation process is illustrated in **Figure 1**.

Figure 1. Evaluation Process



The worth criteria are specified as metrics (e.g., worth criteria which are measurable either objectively or subjectively), which are commonly labeled as measures of merit, measures of effectiveness, measures of efficiency, measures of performance, and so forth. Measures Of Merit (MOM) are often the worth criteria associated with the system as a whole, while Measures Of Performance (MOP) are worth criteria often associated with the system's subsystems (which may descend to the n^{th} subsystem level of the system). Using these labels, the MOP are below the MOM in a hierarchy of worth criteria, with the MOM comprising the MOP and being a function of the MOP values. For example, a robot's MOP may be "Time Efficiency," and this, along with other MOP, then compose the MOM "Vehicle Efficiency."

The objectives, for example, may be to demonstrate the intelligent and cooperative behavior on the part of multiple autonomous robots or robotic vehicles in the context of scenarios relevant to a class of military missions. The primary objectives of achieving (1) intelligent behavior, and (2) cooperative behavior, lead to the definition of worth criteria focused on two system levels: (1) the individual robot or robotic vehicle as a system, and (2) the group of robots or robotic vehicles as a system (i.e., a *system of systems*).

2.1 Procedural Difficulties

The evaluation procedure is a formal procedure, as opposed to an observer's using purely subjective judgment and intuition to pronounce the performance of the system to be a success or failure. However, formal decision-making procedures do not preclude (and often require) the use of subjective judgment. Subjective judgment must be used in developing worth criteria and assigning them to the various performance consequences, as well as in deriving relative weights for the worth criteria (i.e., trading off worth among the various criteria). But if subjective judgment is made explicit and logically consistent, then it can be examined and questioned by all interested parties. The result is more likely to be free of incorrect or poorly formulated assumptions.

Difficulties with the procedure include mapping from one to many from the set of behaviors to the set of metrics, i.e., relating a single performance consequence to several worth criteria. For example, if a robotic vehicle correctly senses and notes an enemy mine, this event could be relevant to the vehicle metrics for Intelligence and Effectiveness. Conversely, there can be a mapping from many to one, i.e., many performance criteria may be related to a single worth criteria. For example, behavior such as finding mines, avoiding rocks, and finding survivors, among others, contribute to the vehicle's Effectiveness.

Other difficulties include the existence of complex patterns of interaction among various aspects of performance and complex patterns of interdependence among subjective notions of worth (such as distinguishing among Intelligence, Effectiveness, and Efficiency; or among Command & Control, Communications, and Effectiveness. It can be difficult to distinguish between interactions among performance consequences (i.e., system behavior), which is a result of physical phenomena, and interdependence among worth criteria imposed by the analyst, which is a result of psychological phenomena. Nevertheless, an evaluation process that combines explicit subjectivity with objectivity is usually better than an evaluation process employing only implicit subjectivity. But "any assessment procedure, to generate comprehensible results, must stipulate very clearly whose point of view is being taken and whose values are to prevail," [28].

2.2 Worth

Underlying the evaluation procedure is the concept of worth, which may be defined as the "conscious perceptions held by an individual relating to his underlying feelings of preference, aversion, and indifference. This includes not only direct awareness of the feelings themselves, but also the entire range of cognitive elements supporting such feelings. Conscious rationalizations, justifications, and explanations would all be included in the meaning of worth," [28]. Worth is a function of an object, the situation in which the object is placed, and the person evaluating the object. Notions of worth are formulated by people observing external objects and they may be projected onto those objects; but worth remains in the subjective minds of the observers. Worth judgements are neither true nor false; they exist in-the minds of human beings.

Ideally, the metrics for intelligent systems should have certain properties. They should be *complete and exhaustive* in that all important performance objectives should be represented by the list of measures. They should be *mutually exclusive* in that no listed measure should encompass any other measure. The metrics should be restricted to performance objectives of the *highest importance*, derivable from lower criteria in a worth hierarchy. They should be relatively *independent* in that decision-makers should be willing to obtain additional satisfaction on one measure in exchange for reduced satisfaction on another measure at a rate relatively independent of the level of satisfaction already attained on each.

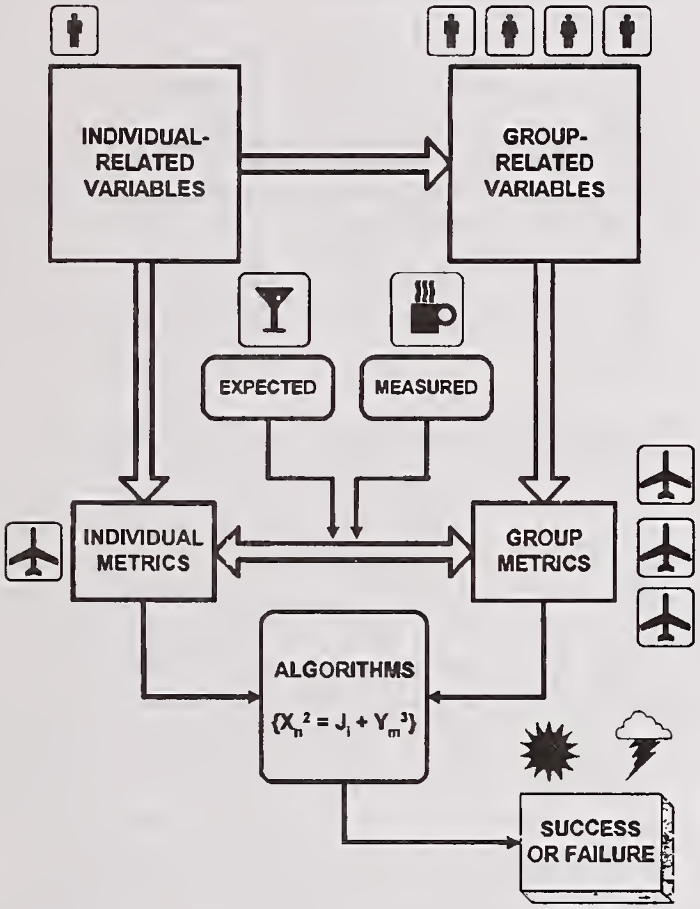
The example metrics selected herein for the intelligent systems intersect somewhat and are therefore *not completely mutually exclusive*, but their exclusivity is sufficient to provide a reasonable evaluation of system performance.

The lowest level criteria in a worth hierarchy should be represented by a simple performance measure. This connects the criteria hierarchy, which emanates from the subjective minds of the decision-makers, with the outer world of physical "reality." For example, the "Number of Targets Detected Per Unit Time" would be a lowest level worth criterion for the higher level criterion "Time Efficiency," which, in turn, would contribute to the evaluation for the higher level worth criterion "Mission Efficiency." The weighted worth scores may be aggregated to calculate an overall index of worth, i.e., an overall determination of success or failure for the intelligent system.

2.3 Variables

The variables and their relationships symbolically represent the operation of the intelligent system, in the context of the environment, in computer simulations or field exercises of missions for the intelligent system. **Figure 2** shows the relationships among the variables and the metrics. Some (although not all) of the system variables are relevant to the mission and group variables. Each of these sets of variables are aggregated, through the application of various algorithms, into metrics; these, in turn, are aggregated, through the application of more algorithms, into a scoring of success or failure. The values of the metrics, i.e., their quantification as a result of simulation or field exercises, determine the success or failure of the exercise of the system (against a priori criteria). In each case, the expected values (e.g., the martini glass in the figure) are compared with the measured values (e.g., the coffee cup in the figure) for the individual and group metrics and submetrics. Success or failure (e.g., of the mission) can depend on the individual, the group, or both.

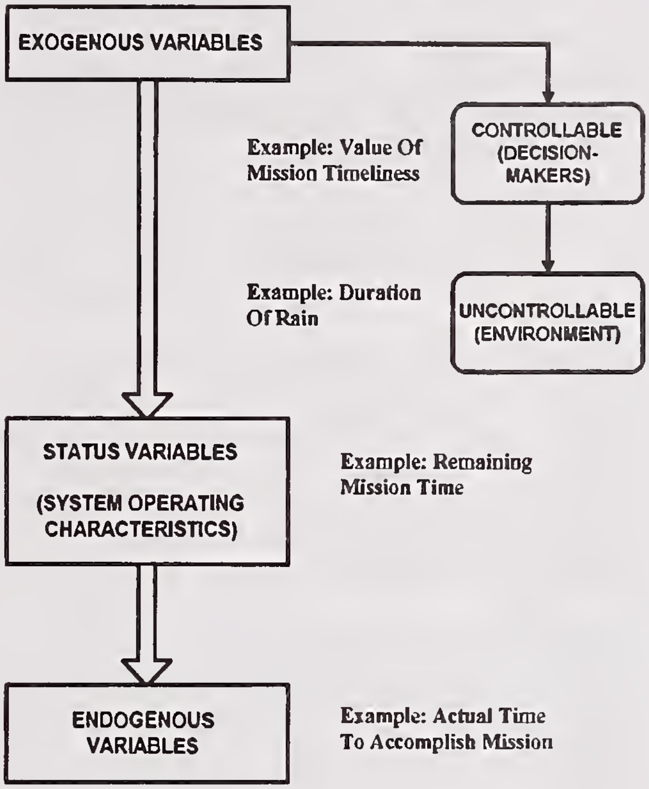
Figure 2. Variables And Metrics



A convenient taxonomy for the intelligent system variables is illustrated in **Figure 3**. Exogenous variables are independent, or input, variables which are generally predetermined and independent of the system. They act on the system but are not acted on by the system. Exogenous variables may be either controllable or non-controllable. Controllable (or instrumental) exogenous variables can be controlled or manipulated by the decision-makers of the system. Non-controllable exogenous variables are generated by the environment in which the system exists and behaves (and not by the system itself or its decision makers). For example, the value of "Mission Timeliness" is a controllable variable, while "Duration (Time) Of Rain" is a non-controllable variable. Non-controllable variables are associated with the individual level; there are none at the mission/group level.

The status variables describe the state of the system. They interact with both exogenous and endogenous variables according to the functional relationships of the system. The value of the status variable may depend on an exogenous or endogenous variable in a preceding time period; when the input is from a portion of a variable's own output from a previous period, a feedback loop exists. "Remaining Mission Time" and "Number Of Objectives Achieved" are examples of status variables.

Figure 3. Types Of Variables

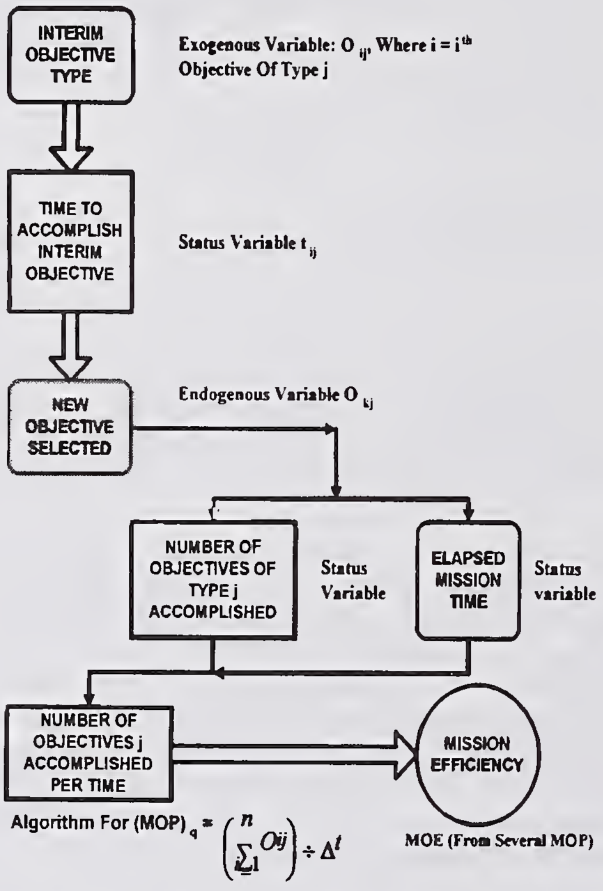

EXOGENOUS VARIABLES
Example: Value Of Mission Timeliness
CONTROLLABLE (DECISION-MAKERS)
Example: Duration Of Rain
UNCONTROLLABLE (ENVIRONMENT)
STATUS VARIABLES
(SYSTEM OPERATING CHARACTERISTICS)
Example: Remaining Mission Time
ENDOGENOUS VARIABLES
Example: Actual Time To Accomplish Mission

Endogenous variables are dependent, or output, variables of the system, generated from the interaction of the system's exogenous and status variables according to the system's operating characteristics. The "Actual Time To Accomplish A Mission" is an example of an endogenous variable.

Whether a particular variable is an exogenous, status or endogenous variable depends on the purpose or nature of the system's processes. For example, "Target Location" may be an exogenous variable if it is specified to the group a priori (as for a fixed target); it may be a status variable if, as a relative location, it is periodically updated as the group moves; and it may be an endogenous variable if it is computed by the group on the basis of sensor inputs.

An example of the use of the variables to derive metrics is given in **Figure 4**. Variables of different types are combined by using an algorithm to obtain a measure of performance: the exogenous variable "Interim Objective Type" (such as a rendezvous point); the status variable "Time Of Interim Objective Accomplishment;" endogenous variable "New Objective Selected" (by the leader vehicle); the status variable "Number Of Objectives" (of this type accomplished); and the status variable "Elapsed Mission Time." The MOP formed from these variables is the "Number Of Objectives Of Type j (such as rendezvous points) Accomplished (by the group) Per Unit Time." The algorithm in this example is simply the sum of the objectives accomplished divided by the mission time. This MOP, along with others (such as "Energy Expended Per Objective Accomplished"), might be combined into a top level metric called "Mission Efficiency."

Figure 4. Example: Variables Transformed To Metrics


INTERIM OBJECTIVE TYPE
Exogenous Variable: O_{ij}, Where i = ith Objective Of Type j
TIME TO ACCOMPLISH INTERIM OBJECTIVE
Status Variable t_{ij}
NEW OBJECTIVE SELECTED
Endogenous Variable O_{kj}
NUMBER OF OBJECTIVES OF TYPE j ACCOMPLISHED
Status Variable
ELAPSED MISSION TIME
Status variable
NUMBER OF OBJECTIVES j ACCOMPLISHED PER TIME
MISSION EFFICIENCY
Algorithm For $(MOP)_q = \left(\sum_1^n Oij\right) \div \Delta^t$
MOE (From Several MOP)

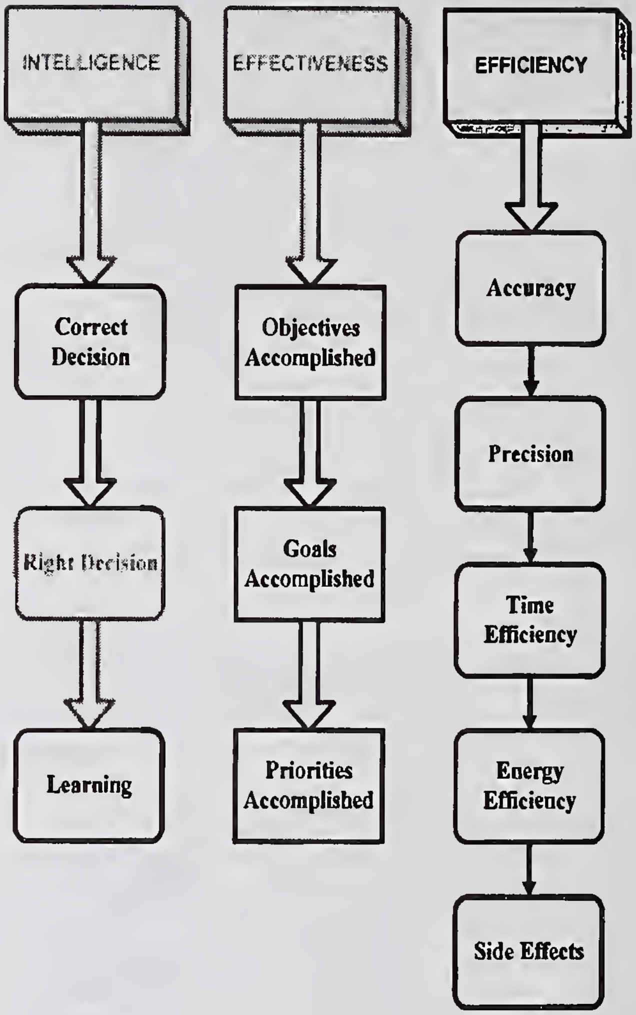
2.4 Metrics For Individuals And The Collective

We define six metrics for intelligent systems for this example. The metrics are not completely mutually exclusive. But they do emphasize three behavioral aspects of an autonomous intelligent system and three behavioral aspects of a group of such systems. Taken together, they provide a summary quantification of how successfully the individual and the aggregate - the collective - perform in the context of their environment and mission.

2.4.1 Individual Metrics And MOP

The three selected top-level metrics are: Intelligence, Efficiency, and Effectiveness, as shown in Figure 5.

Figure 5. Metrics And Submetrics For Individuals



Intelligence is defined here functionally as the ability of the system to make an appropriate choice - an appropriate decision. Because a value is the relative worth of a thing, the basis upon one makes a choice, intelligence is related to values; what is "appropriate" is situation-dependent. In the case of intelligent systems for military-type missions, appropriate choices are those that contribute to the success of the mission, or are perceived by the system to contribute to the success of the mission in the context of the information it possesses.

Information provided by sensors and processed by an intelligent control system can alter the intelligent machine's world model - and learning occurs. The ability to learn, based on experience, is one metric for intelligence. There are two kinds of acceptable decisions that the intelligent system can make: "correct" and "right." A correct decision is the optimum decision the system can make given a meta-view or complete knowledge (ground truth or the "god's-eye" view). The right decision is the optimum decision the vehicle can make given its "real" and limited knowledge. The intelligent machine (or a person) may do well in making correct decisions despite limited knowledge; this kind of decision-making is a metric that evaluates performance in an absolute frame of reference. It is difficult or impossible for mortals to acquire the "god's-eye" view in real life, but it is possible to have such a view in limited scenarios and to evaluate the performance of men or machines against such a standard.

In **Figure 5** the metric "Intelligence" is decomposed into the submetrics (or MOP): "Correct Decision," "Right Decision," and "Learning." "Correct Decision" evaluates the machine's intelligence against *absolute* performance standards. "Right Decision" evaluates the machine's intelligence against a *relative* standard which discounts the limitations of the machine's sensors and world model. The "Learning" MOP measures the ability of the vehicle to adapt to its environment, without necessarily making a decision that leads immediately to altered *external* behavior.

For example, an autonomous robotic vehicle might sense a terrain feature it that doesn't appear in the terrain map stored in its world model. Appropriate learning would occur if the vehicle were to alter its terrain map to include the feature; the vehicle need not have altered its path or motion in order to indicate learning - the change in the world model would be sufficient to indicate learning. If the vehicle were to select a path to its destination that complied with all of its mission criteria, but was then ambushed and destroyed by a hidden enemy about which it could not have known, the vehicle would have made a right decision in its path selection, but not a correct decision.

The metric "Effectiveness" in **Figure 5** is decomposed into the submetrics or MOP: "Objectives Accomplished," "Goals Accomplished," and "Priorities Accomplished." "Effectiveness" is the "bottom line" measure of merit, the measure of whether the mission goal and its interim objectives were achieved by the vehicle. Ordinarily, this might be the main metric, the one with the greatest importance. However, developmental or prototype systems may have, for example, various mechanical-type subsystems that are not of operational quality. It is not absolutely critical to the development of intelligent systems that prototype robotic vehicles accomplish its goals and objectives with overwhelming panache. The display of intelligence is more important in a Phase I effort than the success or failure of the mission - which may depend on the success or failure of a prosaic propulsion system. In the end, of course, with a fielded system, "Effectiveness" is a key metric. Ineffective intelligence is barren, in machines or people.

The tactical "Objectives Accomplished" is an MOP based on the intermediate objectives the intelligent machine is assigned to accomplish on its way to the ultimate mission goal, which accomplishment is accounted for in the MOP "Goals Accomplished." The final MOP for "Effectiveness" is the determination of the "Priorities Accomplished." The priorities are those set in the value-driven logic of the robotic platform, i.e., the relative importance of survival, energy conservation, timeliness, etc. The robotic platform may be able to accomplish most of its intermediate objectives, yet fail at its ultimate goal (just like people often do), or it may achieve its ultimate goal while failing at its intermediate objectives (e.g., getting the lucky break). Also, it may maintain or scramble its priorities while succeeding or failing at accomplishing its objectives and goal. The MOP for "Effectiveness" are thus sufficiently mutually exclusive to highlight different aspects of the robotic vehicle's behavioral and mission performance.

The final metric, that of "Efficiency," is the least important in a development program because a prototype platform's mechanical performance is likely to be inferior to that required for an operational platform. However, it is reasonable to account for this behavior in the testbed and include it in the final metric score. For an operational system, "Efficiency" becomes more important, but not usually as important as "Effectiveness."

"Efficiency" is a measure of how well the intelligent system performs while attempting to accomplish its objectives and goal, and how well it conserves resources. "Effectiveness" measures the ability to accomplish the objectives and goal assigned by the mission. The vehicle (like a person) may be extremely efficient and yet completely ineffective (such as working economically toward the wrong goal); or it may be inefficient, yet able to accomplish its objectives and goal. "Effectiveness" and "Efficiency" are not completely independent, but they are sufficiently different to characterize different aspects of an intelligent system.

There are four MOPs for the metric "Efficiency," as shown in **Figure 5**: "Accuracy," "Precision," "Time Efficiency," "Energy Efficiency," and "Unexpected Adverse Side Effects."

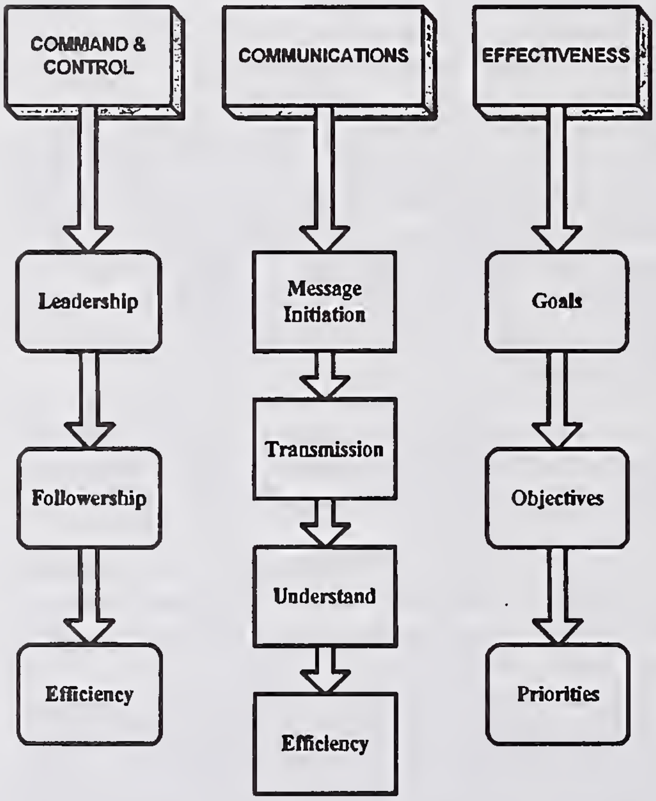
"Accuracy" refers to the robot's ability to achieve its desired states (position, speed, etc.) without significant systematic errors. "Precision" refers to the vehicle's ability to achieve its desired states without significant random errors. "Time Efficiency" measures the accomplishment of objectives and goal per unit time (such as the number of targets detected per minute, or the number of survivors retrieved per hour, the area searched per hour, etc.). "Energy Efficiency" likewise measures the accomplishment of objectives and goal per unit of energy expended (such as the number of mines detected per joule, etc.). The "Unexpected Adverse Side Effects" refer to adverse behavior displayed by the robot due to bugs, glitches, or errors in the vehicle. Such behavior may not prevent the vehicle from accomplishing its mission (or even detract much from its accuracy or precision), but it could reduce efficiency. For example, every 100 meters the robot might inexplicably stop for ten seconds; or it might mistake a wall for an entranceway and try to enter.

Accuracy and Precision are basic to efficient performance and should be weighted somewhat higher than Time and Energy Efficiency. Side Effects, while disturbing and potentially harmful to the success of the mission (or the continuation of a development program itself) is not of high importance in a Phase I development effort; the causes of eccentric behavior presumably can be found and corrected. The existence of peculiar vehicle bugs will become more worrisome as intelligent machines become operational.

2.4.2 Group Metrics And MOP

The three metrics selected for the group or mission level are: "Command and Control" (C^2), "Communications," and "Effectiveness," as shown in **Figure 6**.

Figure 6. Metrics And Submetrics For Groups



"Command and Control" (taken as a single measure) refer to the ability of the robots to exhibit cooperative behavior within a leadership structure. One major system attribute - intelligence - is measured from individual robot or platform behavior. The other major system attribute - cooperation - requires more than one platform for measurement; it is measured from the interactions of multiple intelligent systems. We assume an explicit means of achieving robotic group behavior for the applications of interest (e.g., leader-follower architecture), rather than implicit means (e.g., eusocial architecture).

The multiple systems can be designed to interact in many different ways, just as people in various societies and institutions organize themselves in different ways. In particular, the organizational forms needed to achieve organizational goals can range over a spectrum of types, from collegial to democratic to autocratic to - and so on. The organizational form selected, for example, may be military-autocratic where some robots are leaders and others subordinates, all in a hierarchy of authority and power. (Authority is the *right* to act while power is the *ability* to act).

In **Figure 6** the metric "Command and Control" is decomposed into the MOP: "Leadership," "Followership," and "C^2 Efficiency." While *cooperation* may seem too weak a characterization for the relationship between a military leader and his (its) subordinates, leadership always involves some form of cooperation from followers - even from those under duress.

The definition of "leadership," like that of "intelligence," is vague. Some of the definitions of leadership include [29]:

- □ "Leadership is the exercise of authority and the making of decisions," (Dubin, 1951);
- □ "Leadership is the initiation of acts that result in a consistent pattern of group interaction directed toward the solution of mutual problems," (Hemphill, 1954);
- □ "The leader is one who succeeds in getting others to follow him," (Crowly, 1928);
- □ "Leadership is the process of influencing group activities toward goal setting and goal achievement," (Stogdill, 1948);
- □ Leadership is "the ability to handle men so as to achieve the most with the least friction and greatest cooperation," (Munson, 1921);
- □ Leadership is "the process by which an agent induces a subordinate to behave in a desired manner," (Bennis, 1959);
- □ Leadership is "the activity of persuading people to cooperate in the achievement of a common objective," (Koontz and O'Donnell, 1955).

An ideal form of leadership might be to motivate others such that they perceive themselves to be self-motivated, an invisible, unobtrusive form of leadership. Then, there is the eusocial leaderless leadership, a commonality of purpose arising from the dynamics of group interactions, as exhibited by ants. Unlike human organizations, robotic systems might well be able to accomplish invisible or leaderless leadership.

Effective leadership can be measured by how well the leader's group performs its assigned functions in terms of group productivity and group satisfaction, although in the case of the robotic collective, group satisfaction is not a concern. In human organizations, the effective leader possesses power which originates from his position, from higher authority, and from his traits, abilities and behaviors. The followers of the leader also have traits, abilities and behaviors which contribute to the successful accomplishment of the mission. Between the leader and the followers are their relationships and the task structure. Technology impacts on the triad of leader, followers, and their relationship in various ways; communications technology, for example, can alter the leader's power or facilitate orders to subordinates.

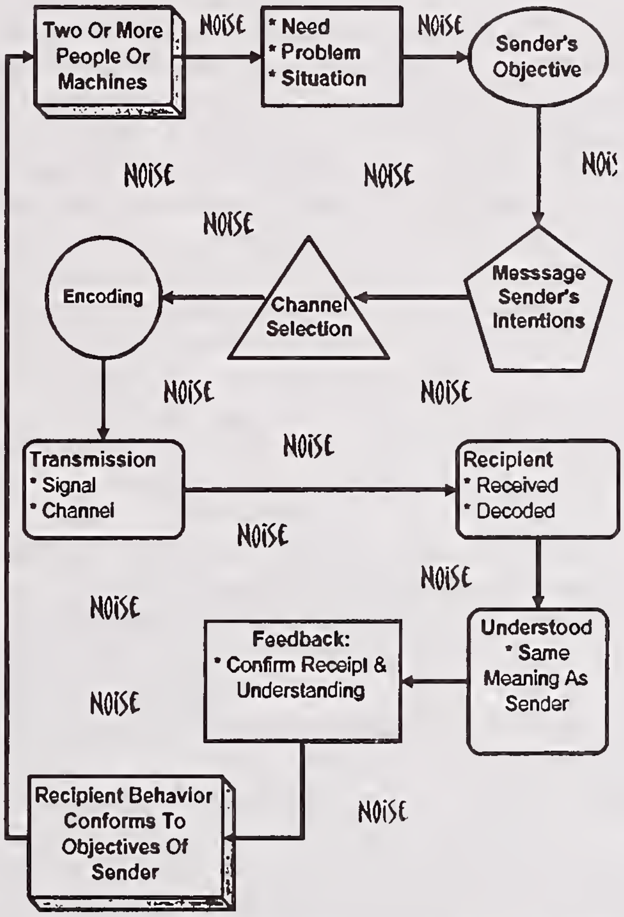
For a robotic collective, there will also be leadership potential (programmed algorithms), behavior (decisions based on value-driven logic), leader-follower relations (inter-vehicle protocols), and task structure (the degree of control and the tradeoff of centralization versus decentralization). The leader robot will take the initiative in making decisions, select tactics and maneuvers, and issue appropriate commands to the follower robot.

There can be no leader without a follower, and there can be no leadership without followership. So, for example, the followership of one robot vehicle will help define the leadership of another robotic vehicle. The subordinate vehicle will respond to commands appropriately, providing feedback to the command vehicle and behaving with restraint (an aspect of control). The subordinate vehicle will take command, transforming itself into the leader vehicle, when the latter cannot function properly because it has been damaged or destroyed. (In the case of two vehicles, the surviving follower becomes the "leader" in the nominal sense of performing the mission tasks of the leader vehicle without leading a subordinate).

The link between the leader and followers is achieved through communication. **Figure 7** illustrates the communication process. One of at least two or more people or machines perceive a need, problem, or situation that requires the transmission of information. The communication initiator - the sender - has an objective in sending the information. The sender formulates a message that contains the information reflecting the intentions of the sender. (Information, in a quantitative context, is a measure of one's freedom of choice when one selects a message from an available set. In this *entropic* view, the message that water is wet, to one who knows this fact, would not contain information. If information is related to choice, and values are the bases of choice, and intelligence is the ability to make appropriate choices, then information, values, and intelligence are related.) The sender selects a channel or medium over which to send the message, encodes the message into the appropriate language and format, and transmits the message over the channel. The recipient of the message receives and decodes it..

For communication to have taken place, the recipient must understand the message, i.e., must extract the information the sender intended. Ideally, the recipient provides feedback to the sender so that the latter knows that the information has been received and understood. Sometimes the feedback consists of the sender's observing subsequent behavior of the recipient that conforms to the objectives of the sender.

Figure 7. Communications



Any step of the communication process can be disrupted by noise. Noise may originate in the sender and disrupt the internal formulation or encoding of the message, or it may originate in the environment and disrupt the transmission of the message, or it may arise in the recipient and discombobulate the decoding or understanding of the message.

The communication process, as outlined **Figure 7**, is true for communication between people (using verbal, written, and other means) or robotic vehicles (using radio frequency, acoustic, optical, and other means).

The “Communication” metric is decomposed into four MOP: “Message Initiation”, “Message Transmission”, “Message Efficiency”, and “Message Understanding”. The MOP correspond to the communications process as outlined.

The initiation and understanding of messages are more important in a Phase I development effort than the performance of the transmission mechanism (radio frequency or acoustic), or the efficiency of the message protocol (length and number of messages needed to convey a quantity of information). Acceptable performance might consist of appropriate messages being initiated and a high percentage understood, given reception (but many messages might not be received to due noise or inadequacies in the transmission system).

The metric "Effectiveness" in **Figure 6** is decomposed into the MOP: "Objectives Accomplished," "Goals Accomplished," and "Priorities Accomplished." Effectiveness, as noted previously, is the *bottom line* metric, the measure of whether the mission goal and its interim objectives were achieved by the robotic vehicles. Ordinarily, this might be the main metric, the one with the greatest importance. However, a developmental robot may be a testbed with mechanical systems that are not of operational quality. As we mentioned previously, it is not absolutely critical to the success of a development program that the leader and follower vehicles accomplish their goals and objectives with overwhelming panache. The display of C^2 ability at the mission/group level (and intelligence at the individual level) is a more important accomplishment during development than the success or failure of the mission.

The tactical "Objectives Accomplished" is an MOP based on the intermediate objectives the robotic vehicles are assigned to accomplish on their way to the ultimate mission goal, which accomplishment is accounted for in the MOP "Goals Accomplished." The final MOP for "Effectiveness" is the determination of the "Priorities Accomplished." The priorities are those set in the value-driven logic (e.g., the relative importance of survival, energy conservation, timeliness, etc.).

The robotic vehicles may be able to accomplish most of their intermediate objectives, yet fail at their ultimate goal (just like people), or they may achieve their ultimate goal while failing at their intermediate objectives. Also, they may maintain or scramble their priorities while succeeding or failing at accomplishing their objectives and goal. The MOP for Effectiveness are thus sufficiently mutually exclusive to highlight different aspects of the leader-follower behavior and mission performance.

2.5 Metrics And The Analytic Hierarchy Process

There are multi-criteria decision-making techniques which can be used to define and weight metrics and evaluate alternative systems and technology for prospective intelligent robots. One such technique, the Analytic Hierarchy Process (AHP), is gaining popularity in the defense community (U.S. and Canada) for aiding in the evaluation of weapons systems, and there are more than 600 papers and books describing the theory and applications of the AHP. The mathematics underlying the AHP is largely matrix algebra wherein one solves for certain eigenvalues [30, 31, 32].

Making decisions about complex problems involving conflicting criteria and several alternatives is not a simple process. Psychological research has demonstrated that the human mind is limited in the number of items it can store in short-term memory. The AHP enables the decision-maker to transcend such limitations by visually structuring a complex problem in the form of a hierarchy. Each factor and alternative can be identified and evaluated with respect to other related factors. The AHP makes it possible to look at the elements of a problem in isolation: one element compared against another with respect to a single criterion. The decision process reduced to its simplest terms - pairwise comparisons. This ability to structure a complex problem, and then focus attention on individual components, improves decision-making. All judgements are synthesized into a unified whole in which the alternatives are clearly prioritized from best to worst.

For example, one might look at two robots and note (quantitatively) that the first weighs more than the second. In addition to observing this, we have an ability (subjectively) to say that the first robot is much more flexible (i.e., has an ability to perform more or varied functions) than the second, or just moderately more flexible, or that the flexibility of the two robots is the same. Or we might quantify the flexibility in terms of a measurable quantity (such as the number of defined functions performed), for example. A multiplicity of such pairwise comparisons of alternatives (or the use of objective data, where available), against various criteria, build a metric that can be used to make judgments or decisions that are more objective and rational than they would be otherwise.

We first performed this kind of analysis for determining robotic "IQ" for autonomous underwater vehicles in 1985 [32]. This work was updated for robotic ground vehicles in 2000 [33]. The results of this analysis is summarized below.

2.6 Example Analysis

As an example from longer lists [33], *exogenous* variables for *individual* robots include: coordinates (starting and final); maximum detection range (passive and active); terrain profile; object (size, speed, acceleration, coordinates; rendezvous coordinates; etc. Sample *status* variables include: vehicle speed (linear and angular); vehicle position; vehicle bearing; sensor status; power status; etc. Sample *endogenous* variables include: probability of bring detected (actively and passively); risk of known and unknown sensors along path; estimated path length; computed position of object sensed actively; computed object speed; etc.

Example *exogenous* variables for robot *groups* include: mission type; mission values; desired vehicle spacing; designated group leader; primary mission objective types (defenses, targets, vehicles, etc.); abort criteria; group clock standard; etc. Sample group *status* variables include: groups destroyed; vehicles per group destroyed; vehicles absent from rendezvous; elapsed/remaining mission time; etc. Sample group *endogenous* variables include: risk of active detection fo group; risk of passive detection for group; number of objects of each type sensed by group; best computed position of object sensed actively by group; etc.

For the AHP Goal to "Evaluate *Individual* Robot IQ," the values of the weights for the metrics and submetrics, previously described, were calculated with the following results:

* Intelligence = 0.54
□ Correct Decision = 0.10
□ Right Decision = 0.27
□ Learning = 0.17

* Effectiveness = 0.30
□ Objectives Accomplished = 0.15
□ Goals Accomplished = 0.06
□ Priorities Accomplished = 0.09

* Efficiency = 0.16
□ Accuracy = 0.05
□ Precision = 0.05
□ Time Efficiency = 0.03
□ Energy Efficiency = 0.02
□ Side Effects = 0.01

For the AHP Goal to "Evaluate *Group* Robot IQ," the values of the weights for the metrics and submetrics, previously described, were calculated with the following results:

* Command & Control = 0.54
□ Leadership = 0.23
□ Followership = 0.23
□ Efficiency = 0.08

* Communications = 0.16
□ Message Initiation = 0.06
□ Message Transmission = 0.03
□ Message Understanding = 0.06
□ Efficiency = 0.01

* Effectiveness = 0.3
□ Goals Accomplished = 0.06
□ Objectives Accomplished = 0.15
□ Priorities Accomplished = 0.09

While there are many ways to evaluate the "IQ" of a robot and groups of robots, a simple (vector) method is to add the products of the values obtained for the individual and group metrics and their associated weights:

[1] Total Score = $\sum_i W_i M_i$

Where W = i^{th} Weight
M = i^{th} Measure (Score)

The scores of each metric are obtained from measuring the submetrics or MOPs in a series of experiments, in a simulation or in the field. Each individual and group metric requires a defined process for obtaining its score, which is then aggregated into the Total Score (or "IQ"). There are many possible approaches or algorithms, an examples are given in [33]. For example, to evaluate the group Communications metric one might define:

[2] $SC = \sum_{i=1}^{3} W_i R_i + W_4 E$

Where:

* SC = Score For Communications
* R_1 = NMI/TMI = Message Initiation Ratio
* R_2 = NESR/TMI = Transmission Ratio
* R_3 = NMU/NESR = Understanding Ratio
* NMI = Number Of Right Messages Initiated
* TMI = Total Number Of Messages Initiated
* NESR = No. Messages Actually Encoded, Sent, And Received
* NMU = No. Of Messages Rightly Understood By Recipient
* W_i = Weight of i^{th} MOP (As Previously Calculated)
* E = Evaluation Of Message Lengths And Quantity Compared With What Would Be Right: $(0 \leq E \leq 1)$

Example steps to measure the MOP associated with group Communication include:

Step 1: Store the time of initiation of messages (i.e., a new plan of a robot to send a message to another robot), the contents of the messages, the time of transmission of the messages, the time of reception of the messages, and the contents of the messages as received by the receiving robot.

Step 2: The analyst, after the mission, calculates the Message Initiation Ratio, Transmission Ratio, and Understanding Ratio. The analyst judges the rightness of the message contents, as well as the rightness of the understanding of the messages on the part of the receiving robot, based on the robot's subsequent behavior. The analyst also judges the rightness of the message lengths and quantity (too much or too few) of messages and scores this as previously described.

Step 3: The analyst weights and combines the scores of the four MOP associated with group Communications to calculate the Communications Score, and weights and combines this score with the other weighted metric scores to obtain a final value for the group "IQ."

Another example is a method for scoring the Effectiveness metric for the individual robot. To score the accomplishment of the tactical objectives and goal of a mission, the human evaluator notes the number of objects (e.g., survivors in an urban search and rescue operation) to be sensed or acted upon (e.g., located, given water or oxygen, carried to safety) by the robot and divides by the total number of such objects in the scenario. A similar ratio is taken for the number of positions (rendezvous locations, assigned reconnaissance positions, etc.) the vehicle should have visited. The Priority MOP is evaluated by determine whether the priorities in the value-driven logic were followed as assigned, or modified according to the rules, through the mission. For example, a score (e.g., 0 to 4) can be assigned to each priority, then they are summed and averaged. For Effectiveness we then have:

[3] $R(O_{ij}) = (\sum_{j}^{m} \sum_{i}^{n} O_{ij}) / O_T$

[4] $R(P_i) = (\sum_{i}^{m} P_i) / P_T$

[5] $S(\mathrm{Pr}) = \sum_{i=1}^{4} \mathrm{Pr}_i / 4$

Where $[0 \leq \mathrm{Pr}_i \leq 4]$

Where:
* $R(O_{ij})$ = Object ratio (for goal or objectives)
* $R(P_i)$ = Position Ratio (for goal or objectives)
* S(Pr) = Priority Sum Average
* O_{ij} = The i^th^ Object of Type j (For example, j=1=survivor; j=2=mine; j=3=areas to be avoided; etc.)
* P_i = The ith position (goal or objective) Visited
* m = Total Types of Objects Sensed or Acted Upon (Or Total Position Visited)
* n = Total Objects Of Each Type Sensed Or Acted Upon
* O_T = Total Number Of Objects Robot Should Interact With To Achieve Goal Or Objectives
* P_T = Total Number Of Positions (Goal Or Objectives)
* Pr_1 = Stealth
* Pr_2 = Survival
* Pr_3 = Timeliness
* Pr_4 = Energy

The steps to measuring robot "Effectiveness" are:

Step 1: Specify the tactical plan for the mission. In an urban search and rescue mission, for example, this might be to: Search for a specified object or person; perform Reconnaissance (to search for entrances or signs of life); perform Surveillance (in a specified region); Map (a specified region); Retrieve a person, etc. If the mission goal for a group of two robots were to locate and retrieve survivors from within a room on an upper floor, the mission goal of one of the robots might be to locate a path to the upper floor by searching a lower floor. The mission-level goal consists, for example, of a state-graph defining a sequence of potential commands that the mission executor will issue to the group level planner. Store the robot's mission goal as specified at the start of the mission, and any changes of the goal made during the mission, with the time of the changes.

Step 2: Store the robot's input tactical commands for decomposed intermediate objectives (if they are changed during the mission, store the changes along with the times of the changes), then store the changes in the state-graph which indicate that a robot's input command has been accomplished by the robot, and note the time of the accomplishment.

Step 3: Determines whether the robot has substantially accomplished its mission goal. Calculate the score quantitatively e.g., using an Object Ratio or Position Ratio (for example, the ratio of entrances to a collapsed building located to the total number of entrances in the building) or qualitatively (assign a score to the mission).

Step 4: The Object Ratio and Position Ratio are used by the analyst to calculate the Objective Score, summing the number of objects or positions that the robot interacted with in the accomplishment of its intermediate objectives and dividing by the total number of such objects or positions with which it should have interacted (according to ground-truth).

Step 5: The values used in the value-driven route planner, such as for stealth, survival, timeliness, and energy, should be stored for retrieval by the analyst. At the conclusion of the mission, the analyst calculates the Priority Sum Average by evaluating the behavior the vehicle, assigning a scores, and taking an average.

Step 6: The analyst weights and combines the scores of the three MOP associated with vehicle Effectiveness (i.e., Goals, Objectives, and Priorities) and calculates a total score for "Effectiveness."

3.0 Acknowledgments

We gratefully acknowledge that this work was funded in part by Contract 43NANB009812 from NIST.

4.0 References

[1] Conant, Roger, ed. *Mechanisms Of Intelligence: Ross Ashby's Writings On Cybernetics*. Seaside, CA: Interscience Publishing; 1981, p. 178.
[2] *Glossary On Cybernetics And Systems Theory*, American Society For Cybernetics; 1984, pp. 21-22.
[3] Gevarter, William B. *Intelligent Machines*. Englewood Cliffs, NJ: Prentice-Hall, Inc.; 1985, p. 229.
[4] *The Economist*; 17 October 1987, p. 13.
[5] Gregory, Richard L. *The Oxford Companion To The Mind*. Oxford: Oxford University Press, 1987, p. 410.
[6] Chen, Minder, Yihwa Irene Liou, and E. Sue Weber.

Developing Intelligent Organizations: A Context-Based Approach to Individual and Organizational Effectiveness. *Journal of Organizational Computing* 2(2): pp.181-202, 1992.

[7] McCarthy, J., and P. Hayes. Some Philosophical Problems From The Standpoint Of Artificial Intelligence. In *Readings In Artificial Intelligence*, Bonnie Lynn Webber, Nils J. Nilsson, eds. Palo Alto, CA: Tioga Publishing Co 1981, p. 432.

[8] Fischler, Martin A., and Oscar Firschein. *Intelligence: The Eye, The Brain, And The Computer.* Addison-Wesley Publishing Co., Inc.; 1987, p. 4.

[9] Schrage, Michael. Book Renounces The Early Promise Of Artificial Intelligence. *Washington Post*, 1 June 1987, p. 17.

[10] Wooldridge, Dean E. *Mechanical Man: The Physical Basis Of Intelligent Life.* New York: McGraw-Hill Book Co.; 1968, p. 128.

[11] Ashby, W. Ross. *Design For A Brain.* Chapman & Hall; 1976.

[12] Doyle, Jon. *A Model For Deliberation, Action And Introspection,* MIT Artificial Intelligence Laboratory TR-581 May 1980, pp. 15-16.

[13] Barr, Avron, and Edward A. Feigenbaum. *The Handbook Of Artificial Intelligence.* Stanford, CA: Heuristic Press; 1981, p. 147.

[14] Rorty, Richard. *Philosophy And The Mirror Of Nature.* Princeton University Press; 1979, p. 125.

[15] Sowa, J.F. *Conceptual Structures: Information Processing In Mind And Machine.* Addison-Wesley Publishing Co.; 1984, p. 2.

[16] Gregory, op. cit., p. 149.

[17] Gevarter, op. cit., p. 225.

[18] Fischler, op. cit., p. 383.

[19] Minsky, Marvin, and Seymour Papert. *Perceptons.* MIT Press; 1969.

[20] Pugh, George E. *The Biological Origin Of Human Values.* Basić Books Inc.; 1977.

[21] Gregory, op. cit., p. 383.

[22] Koza, John R. *Genetic Programming.* MIT Press; 1992, p. 14.

[23] Stonier, Tom. *Beyond Information: The Natural History of Intelligence.* Springer-Verlag; 1992, p. 70.

[24] Stonier, op. cit., p. 72.

[25] Stonier, op. cit., p. 85-105.

[26] Arkin, Ronald and George Bekey, *Robot Colonies*, Kluwer Academic Publishers, 1997, reprinted from the special issue of the journal of *Autonomous Robots*, Vol. 4, No. 1, March 1997.

[27] Cao, Uny Y., Fukunaga, Alex S., and Andrew B. Kahng, *Cooperative Mobile Robotics: Antecedents And Direction*, in Arkin, Ronald and George Bekey, *Robot Colonies*, Kluwer Academic Publishers, 1997, pp. 7- 27.

[28] Miller, James R., *Professional Decision-Making: A Procedure For Evaluating Complex Alternatives*, Praeger Publishers Inc., 1970.

[29] Finkelstein, Robert, "Combat Robotics: Implications For Leadership," *Unmanned Systems*, Winter 1987, pp. 9-10.

[30] Saaty, Thomas L., *Decision Making For Leaders: The Analytical Hierarchy Process for Decisions in a Complex World*, Wadsworth, Inc., 1982.

[31] Saaty, Thomas L., *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*, McGraw-Hill Book Co. 1980.

[32] Saaty, Thomas L., A Scaling Method for Priorities in Hierarchical Structures, *Journal of Mathematical Psychology*, 15, pp. 234-281 (1977).

[32] Finkelstein, Robert, *Measures Of Performance And Effectiveness For The MAUVE System: MOE Report Number 5*; Robotic Technology Inc., Item Number 0002AE, Contract No. 50SBNB7C4549, National Institute Of Standards And Technology, June 1987.

[33] Finkelstein, Robert, *Metrics: Evaluating The Performance Of Intelligent Systems*; Robotic Technology Inc., Contract No. 43NANB009812, National Institute Of Standards And Technology, June 2000.

5.0 Author Biography

Dr. Robert Finkelstein is an Associate Professor and Director of the MBA Program at the University of Management and Technology and an Adjunct Associate Professor at the University of Maryland. He is President of Robotic Technology Inc. and Chairman of Decide-Now.Com, Inc. He received a DBA in Systems and Cybernetics from the George Washington University (GWU) in 1995; Ap.Sci. in Operations Research from GWU in 1977; M.S. in Operations Research from GWU in 1974; M.S. in Physics from the University of Massachusetts in 1966; B.A. in Physics from Temple University in 1964.

Machine Intelligence Ranking

Paul P. Wang
Duke University, BOX 90291
Department of Electrical and Computer Engineering
Durham, North Carolina, 27708-0291, USA

ABSTRACT

This talk addresses a number of issues which were inspired by the draft of a document on Metric for Intelligence of Constructed Systems. The constructed systems here literally mean an autonomous control system. It is important to note the opinions expressed in this talk reflect the thoughts of the author and they do not reflect on any institution, organization or professional society.

There are six issues being raised in this talk. The first issue deals with the discussion on the role NIST should play. The institution of NIST was chartered to serve American citizens to improve their well being and the noble goal of pursuing a life of happiness. One of the most important tasks is to measure, standardize, and rank the engineering systems and the advancement of the technology objectively. The autonomous constructed systems were singled out with a high profile to reflect their importance. Are there any other man-made systems which are equally or more important?

Second issue has to do with measuring intelligence. We are measuring intelligence because technology embraces intelligence giving us a superior and high performance system. On the other hand, it is not NIST's mission to do all that because it is there! The fundamental issue, however, is to serve the citizens better via improved technology which requires intelligence. The definition of intelligence, however, is no simple matter, as well as the definition of serving citizens. Both cover a wide spectrum of needs and desirable things other than autonomous systems of which intelligence so happen needed to be put in the center of the stage.

The third issue to be raised is the definition of "machine intelligence" and how to measure it? Since the definition of human intelligence is complex and difficult, the definition of machine intelligence is even more difficult!

The fourth issue has to do with the performance evaluation of engineering systems. This issue deals with value judgement. The debate by the citizens among all walks of life and society as a whole must be carried out in order to establish value judgement as a benchmark for measurement, testing, and evaluations.

This brings us to the issue of testing and measuring. The central issue is how are we to conduct the machine intelligence test? It is not a simple matter because we have not yet settled the definition of machine intelligence!

Equally important is the issue of understanding the crux of our present technology and forecasting of future technology. The reason is due to the fact that there is absolutely no unique way to realize a high performance system. Here we are talking about a federal institution to set the standard to evaluate and rank a high performance system. Generally speaking, the position this paper takes is that some of the issues raised in white papers are over simplified. Some of the long term frame works have not been covered adequately. If one believes in the basic assumptions, hypotheses set by the white paper and willing to live with all the constraints already being laid out, then this paper has no validity. The feeling of this author is that the constraints dealing with intelligent machines are overly constrained and a liberation effort hence is needed.

The main concerns are: the basic charter of the institution is unclear, the science on intelligence is too complex, the need of application areas is too complex, and the technologies available are too uncertain to reach a consensus.

With these constraints, I must say that the white paper is truly an outstanding document full of creativity, imagination, and innovative ideas. Congratulations to Alex Meystel and Jim Albus.

Survivability and Competence as Measures of Intelligent Systems

Reid Simmons
Robotics Institute
Carnegie Mellon University

While the workshop is appropriately named "Measuring the Performance of Intelligent Systems", there may be come confusion that the goal is actually about measuring the *intelligence of systems*. While measuring performance is a worthy, albeit difficult goal, I believe that trying to measure intelligence itself is misplaced. To me, it seems pointless to debate whether, for instance, playing chess exhibits more "intelligence" than exploring Mars, or whether using speech is inherently more intelligent than doing object recognition. From both pragmatic and philosophical viewpoints, the more that we can make it clear that we are interested in *performance*, rather than intelligence, *per se*, the better off we will be.

So, what criteria are to be used for measuring the performance of intelligent systems? I think that the two most important characteristics are survivability and competence. By *survivability*, I mean the ability of a system to cope with diversity in the environment, as well as internal faults (hardware and software). By *competence*, I mean the ability of a system to successfully perform tasks. Both survivability and competence can be measured either empirically or formally. Empirically, survivability can be measured by carefully controlling environmental inputs and by modifying the internal state of the system (such as by deliberately causing hardware faults). Formally, with the right model one can quantify the range of environmental conditions and internal states that can be handled successfully. Similarly, one can measure competence either empirically or formally by controlling for the range of tasks and the environments under which those tasks are to be performed.

This, of course, begs the question as to how to set up the experiments in an unbiased and controlled fashion, and how to model tasks and environments so that formal evaluations are possible. Unfortunately, I do not have good answers for those questions, at this time (although we are working on it!). The problem is that most intelligent systems exhibit chaotic behavior - small deviations in input conditions lead to wide deviation of behavior (of course, many intelligent systems are also chaotic in the colloquial sense, but that is another matter...). Thus, it is very difficult to set up "the same" conditions to test different systems. One can never be sure if the results are due to actual differences between the systems themselves, or due to small differences in the environments. While simulation can be used to perform standardized experiments, simulators have the disadvantage that they tend to be rather simple models of reality, and so may not capture the essence of what makes survivability and competence difficult.

What about things like robot competitions and Turing tests? I am all for them, but not as quantitative measures of performance, since they suffer from the problem of variability, as described above. The reason that they are valuable is that they come *close* to standardizing tasks and environments in realistic settings, and so can be used by *developers* of intelligent systems to gauge progress, in qualitative ways, against the state of the art. While it is dangerous to use the results of such competitions to conclude anything about one system vs. another (especially one technology vs. another, such as neural nets vs. expert systems), competitions are useful as a type of "bread-boarding" exercise.

Finally, an important aspect of intelligence is *adaptability.* The question is whether adaptability should (or can) be measured independently from survivability and competence. I would argue that adaptability is merely one way of increasing a system's survivability and competence, and thus should not be considered independently. While it may turn out to be true that adaptable systems are generally more survivable and competent, it seems clear to me that this is a hypothesis that needs to be demonstrated empirically, or proved formally. In the absence of such proof, it seems to make little sense to measure adaptability in isolation.

In summary, survivability and competence are two critically important characteristics of intelligent systems. While it is possible to devise ways of measuring both, in a rigorous fashion, it is difficult due to the fact that autonomous systems interacting with complex environments tend to be chaotic. But, that fact should not lessen our resolve to try and measure performance - it only serves to make us aware of the limitations and difficulties of the enterprise.

Two measures for the "intelligence" of human-interactive robots in contests and in the real world: expressiveness and perceptiveness

Illah R. Nourbakhsh
The Robotics Institute
Carnegie Mellon University

Practical measures of intelligence are generally predicated on a social-anthropocentric view of intelligence. This is hardly surprising, but is undesirable because it results in intelligence testing procedures that are uninformative when the subject is not human. For example, the classical Turing Test measures machine intelligence using the yardstick of human social dialogue, in written form, as its gold standard. The problem is that such methodology is implicitly pass fail. Rather than providing a relative measure for machines that are clearly inferior to humans at social human interaction, this test simply fails all such machines until and unless some superior machine simply passes. In airness, it is possible to mitigate this to a small degree by narrowing the content area of the test.

Nevertheless, the Turing Test as applied to the mobile robot system suffers generally the same fate. One can imagine, for instance, a robot Turing Test in which the human teleoperated robot is compared in performance to an autonomous robot in tasks such as navigation, manipulation and robot-human interaction. But the robot will continue to suffer because its raw percepts and raw effectors are not comparable to that of a human. The solution, to force the teleoperating human to use the same percepts as the robot itself uses, results in a robot that whether teleoperated or not is disappointingly unintelligent even when it successfully passes such a robot Turing Test. The problem, then, is that a robot's potential for interaction imposes an upper bound on its potential for intelligence.

Based on this premise, I will propose in my talk that the form of intelligence about which we care most in the case of autonomous robots is interaction.

I will present a methodology for measuring the potential of a robot to engage in rich interaction, thereby establishing a behavioral and analytical way of measuring intelligence without reverting to a direct anthropocentric pass fail test. I will define the concepts of expressiveness and perceptiveness, which together place both upper bounds and lower bounds on interactivity and thereby intelligence. Expressiveness is a measure of the output richness of an electromechanical system. One can quantify expressiveness in terms of the average effectory branching factor of an agent in its observable output space.

Perceptiveness is a measure of the fidelity of an electromechanical system's effective mapping from environmental change to output. This too can be quantified by computing the set of possible output trajectories of an agent in its perceptual workspace. These two measures prove to be particularly useful because they contain no bias with respect to behavior-based and model-based robot architectures. After defining expressiveness and perceptiveness, I provide some quantitative results comparing the expressiveness and perceptiveness of a simple unicellular organism, the dinoflagellate, to that of several popular mobile robots. These quantitative results demonstrate that from the perspective of interactivity mobile robots have a long way to go before challenging human intelligence.

PART II
RESEARCH PAPERS

3. INTELLIGENCE OF DISTRIBUTED AGENTS

Goodness Of Fit Measures For Intelligent Control Of Interacting Machines

Shashi Phoha
David Friedlander
Applied Research Laboratory
The Pennsylvania State University
University Park, PA 16802
sxp26@psu.edu

Abstract

Technology developments in computing and communications have enabled the development of intelligent machines which change their internal states in response to real time interactions with other machines or smart sensors in an ad-hoc network. Complex systems such as these machines may be used to implement a collaborative surveillance sensor network, a multi-robotic mine hunting mobile network, or command and control of multiple hostile or friendly aircraft in an air campaign.

This paper characterizes and evaluates cognitive response in distributed systems of interacting machines. We present a mathematical model of dynamic evolution of such systems and characterize intelligent behaviors like self-organization, identification of friendly or hostile agents, collaboration for achieving common goals, defensive or offensive action against hostile agents, etc. Exploration of the concept of goodness of fit regarding intelligent behavior in interacting machines relates to (*i*) the contextual performance of the network in the presence of expected perturbations in its operational environment, and (*ii*) the quality of adaptation it provides in order to deal with incorrect or incomplete information and unexpected changes in its operational environment. The former relates to behavioral intelligence in executing assigned tasks in a dynamic environment. It can be evaluated from the perspectives of various users by analyzing system response. The latter relates to more intangible characterizations of intelligence akin to creativity and adaptation. Measures of intelligent global behavior of these networks are formulated in terms of their ability to adapt to unexpected perturbations in the environment and the robustness of their responses. This paper develops measures of fit as the ability of the network to adapt to variations in the operational dynamics of the system. These measures assess the overall intelligence of the system in terms of its goodness of fit evaluation for dealing with variations of the plant model for which the network was designed. The quantification of these measures leads to constructive methods of engineering distributed intelligent systems with specified levels of intelligence.

Key Words: *distributed cognition, extrasensory network intelligence, behavior based control, goodness of fit measures, network adaptation, permissiveness, robustness.*

1.0 Introduction

The inherent complexity of controlling a distributed dynamic system implemented on an ad hoc network of interacting machines stems from the fact that an accurate plant model based on physical laws cannot be easily formulated. Concurrent dynamic processes embedded at each node of the system interact in highly non-linear, time-varying and stochastic ways and are subject to unpredictable environmental disturbances. Hence model-based conventional control techniques are inadequate. Alternate methods of designing controllers whose structure and outputs are determined by empirical evidence through observed input/output behavior, rather than by reference to a plant model, are necessary. Several techniques for such non-linear controller design have recently been proposed in recent literature on Intelligent Control [Harris 94, Levis 93, Albus 93, Ramadge 87, Phoha 92, Phoha 98]. Albus [93] has developed the Real Time Control Architecture in which sensor and processing, value judgment, world modeling and behavior generation subsystems interact to adaptively generate appropriate response behaviors to sensor observations and knowledge of mission goals. Meystel [93] has also proposed a nested hierarchical control architecture for the design of Intelligent Controllers. Brooks' subsumption architecture [Brooks 86] for intelligent control is based on achieving increasing pre-specified levels of competence in an intelligent system by examining outputs of lower levels. In Phoha [92, 00] and Ray [93, 95] we have modeled the essential dynamics of these distributed systems as a network of interacting automata that change their internal states through interactions with other nodes or the environment. Due to the well-known relationship between automata and formal languages [Hopcraft 79], we have thus introduced a hierarchical formal language structure for multi-layered dynamic control [Peluso 96] for mission planning and execution.

This paper develops constructive methods of formal language based modeling and intelligent control of interacting machine networks. Section 2.0 presents the mathematical representation of the network as an interacting automata. Section 3.0 formulates the control analysis and synthesis problems. Section 4.0 develops goodness of fit measures for intelligent network behavior. Section 5.0 discusses a method for constructing intelligent machine networks with pre-specified intelligence to adapt to unmodeled dynamics. Thus, a constructive mechanism is formulated for designing intelligent controllers for distributed networks. Applications and open issues are discussed in Section 6.0.

2.0 Models of Machine Interactions

In order to model the concurrent and interacting behaviors of autonomous agents in the complex dynamic system, we extend the automata representations of individual agents to form an interactive automata network, defined as a pair (*G*, $\{a_i\}$) consisting of a cellular spacc *G*, a potentially countably infinite, locally-finite bi-dirccted graph, and an associated family of interacting automata a_i, allocated to each vertex (cell) of a graph. In particular, a_0 represents the operational environment. Each automaton a_i has a finite d_i number of neighbors to communicate with, and has the form

$a_i = (Q_i, \Sigma_i, \Gamma_i, \delta_i, q_{0i}, Q_{Fi})$ where
Q_i is a finite set of control states representing
values of all variables,
Σ_i is a finite alphabet of input events and
$\Sigma_i = \Sigma_0 \times \Sigma_{i_1} \times \ldots \times \Sigma_{id_i}$,
Γ_i is a finite alphabet of output events and
$\Gamma_i = \Gamma_0 \times \Gamma_{i1} \times \cdots \times \Gamma_{id_i}$,
δ_i is a local transition function $\delta_i : Q_i \times \Sigma_i \rightarrow Q_i \times \Gamma_i$
$q_{0i} \in Q_i$ is an initial control state, and
$Q_{Fi} \subseteq Q_i$ is the sct of terminal control states.

The dynamic system operates locally as follows: an interactive automaton a_i occupies each vertex (cell) *i* of *G*. Asynchronously, each a_i looks up its inputs from neighbors $x_{i1}, \ldots, x_{id_i}$, input from an environment x_{i0} and its own state x_i, and then changes its state and produces outputs for the neighbors and the environment according to a local dynamics δ_i. This atomic move is repcated any (possibly very large) number of times.

The environment is modeled uniformly as an interactive automaton, which can be nondeterministic or stochastic, assuming incomplete knowledge about the environment. A distributed environment is modeled as a subnetwork instead of a single nodc. Thus the local transitions δ_i induce a global evolution from one system configuration to another. This global cvolution of the system is viewed as a self-map

T: C → C, where C = $\prod_i (Q_i \times \Gamma_i)$ are configurations (total states) of the network, such that

$$\mathbf{T}(x)_i = \delta_i (x_i, x_{i_0}, x_{i_1}, \ldots, x_{i_{\delta_i}}).$$

The *orbit* of *x* is the sequence of configurations

$$\{ \mathbf{T}^t(x) \}_{t \geq 0} := x, \mathbf{T}^1(x), \mathbf{T}^2(x), \mathbf{T}^3(x), \ldots$$

resulting from successive iterations of the global rule on *x*. As a dynamical system, the most basic question about a global map **T** is the effect of its repeated application in phase space to a random given configuration *x*.

3.0 Control Analysis and Synthesis Problems

The *forward*, or *control analysis*, problem is the following: given local transition rules that determine the local interaction of each automaton with its neighbors (namely the dynamics δ_i), characterize the global effect of the rules on an arbitrary initial configuration *x*. In other words, determine a specific description of the orbits of arbitrary configurations under **T** to identify their long-term (asymptotic) behavior. The forward problem is also known as the prediction of the emerging global behavior from local rules.

The synthesis problem is the inverse problem, which from the point of view of constructing control laws, is even more important. Given a desired global effect on configurations, determine whether there exist, and if so, find the local rules δ_i whose induced global rule is precisely **T**. These local rules will then yield a parallel algorithm for the underlying parallel interacting automata to solve the problem of computing **T**(*x*) for any configuration *x* in C.

Assume, without loss of generality that $\Sigma_i = \Sigma_j \forall i \neq j$ *and* Σ^n is the n^th^ cross product of Σ. Then for $i = 1,2,\ldots,n$, define $S_i = \{(\Phi,\ldots,\sigma_i,\ldots,\Phi) : \sigma_i \in \Sigma\}$. $\{S_i\}$ is a partition of Σ^n under the assumption that one event can occur at a given instant. Let *L*(*P*) denote the open loop language and ø the null event of the plant generated by (*G*, a_i) i.e. $L(P) \subseteq \Sigma^{n^*}$. $K \subseteq L(P)$ is called a controller of (*G*, a_i) *if* *K* is prefix closed in Σ^{n^*} and $K = \prod_{i=1}^{n} K_i$ where K_i is a controller for the subplant represented by a_i in the Ramadge and Wonham sense [Ramadge 87].

Assume that a weight $w_i > 0$ is associated to S_i such that $\sum_i w_i = 1$. Following [Friedlander 00] we define a measure of a given language $L \subseteq \Sigma^{n^*}$ as follows: $\mu(L) = \sum_{i=1}^{n} w_i \Delta_i(L)$,

where $\Delta_i(L) = 1$
if $\exists S = (S_1, S_2, \ldots S_n) \in L$
such that $\Phi \neq S_i \in S_i$; 0 otherwise.

In the next section we will use the notion of the measure of a plant language to evaluate the intelligence inherent in a controlled plant.

4.0 Goodness of Fit Measures of Intelligent Behavior

Exploration of the concept of goodness of fit regarding intelligent behavior in interacting machines relates to (*i*) the contextual performance of the ad hoc network in the presence of expected perturbations in its operational environment, and (*ii*) the quality of adaptation it provides in order to deal with incorrect or incomplete information and unexpected changes in its operational environment. The former relates to behavioral intelligence in executing assigned tasks in a dynamic environment. It can be evaluated from the perspectives of various users by analyzing system response. The latter relates to more intangible characterizations of intelligence akin to creativity and adaptation. We call this Extrasensory Intelligence and proceed to define it as follows:

4.1 Extrasensory Intelligence

In order to characterize the extrasensory intelligence of a controller *K*, for a plant P modeled as an interacting automata (G, a_i, $i = 1,\dots,n$), we study the set of plants which are controllable by *K*.

A plant P' = (G ,a_i , i = 1,...,n) is said to be a k^{th} generation variation of the plant (G, a_i) *iff*

$$k = \max_i\{k_i : \text{generating graphs of } a_i \text{ and } a_i' \text{ differ in exactly } k_i \text{ edges (d-transitions)}\}.$$

For a definition of a generating graph of an automaton, see [Ramadge 87, Phoha 92].

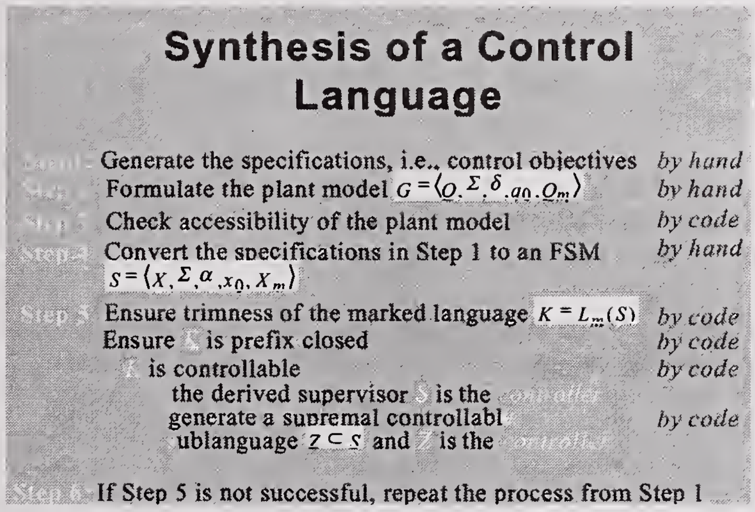
An extrasensory intelligence function for a controller *K* for a plant *P* represented by an interacting automata model [G, a_i ,i=1,...,n] may be defined as the function $I : Z^+$? [0,1] such that $I(k)$ = proportion of plants P' controllable by *K* which are *k* generations apart from its nominal plant model (G, a_i , i = 1,...,n). The objective would be to characterize controllers, *K*, which achieve a specified value of $I(k)$ for a given *k*.

We proceed to define other relevant measures of extrasensory intelligence as follows:
Define

(*i*) $IQ_k(K,P) = \dfrac{\mu[L(P\,|\,K)]}{\mu[L(P)]} \sum_{P_k'} \mu[L(P_k') - L(P)] \cdot D[L(P_k')]$,

where P_k' is any k^{th} generation variation of plant *P*, and $D[L(P_k')] = 1$ if *K* controls P_k', 0 otherwise; and $L(P/K)$ is the measure of the language of the closed-loop, i.e. the controlled plant.

(*ii*) $IQ(K,P) = \lim_{k\to\infty} IQ_k(K,P)$.



Synthesis of a Control Language

Generate the specifications, i.e., control objectives *by hand*
Formulate the plant model $G = \langle Q, \Sigma, \delta, q_0, Q_m \rangle$ *by hand*
Check accessibility of the plant model *by code*
Step 4 Convert the specifications in Step 1 to an FSM $S = \langle X, \Sigma, \alpha, x_0, X_m \rangle$ *by hand*
Step 5 Ensure trimness of the marked language $K = L_m(S)$ *by code*
Ensure is prefix closed *by code*
is controllable *by code*
the derived supervisor is the
generate a supremal controllabl *by code*
ublanguage $Z \subset S$ and is the
Step 6 If Step 5 is not successful, repeat the process from Step 1

Figure 1. Controller Synthesis Process

5.0 Constructing Systems with Prespecified Intelligence

5.1 Building Brains for Interacting Bodies

In this section we examine how controllers can be iteratively designed to achieve prespecified levels of extrasensory intelligence in a complex plant represented by interacting automata (G, a_i, i=1,...,n). We first develop a controller synthesis method which synthesizes a controller for two interacting automata given the specifications for controlling each of the plants represented by the automata. This method can then be iteratively used to construct controllers for other interacting nodes until a controller for the entire plant is designed. Then we formulate the automata models for all the first generation variations of the plant (G, a_i) and attempt to synthesize these controllers using the synthesis automation process described below in 5.2. If a first generation plant variation can be synchronously synthesized, then we have increased the extrasensory intelligence $I(1)$ of the new controller proportionately. Iterate this process until a prespecified value of $I(k)$ is achieved. The entire process can be repeated from here on to achieve prespecified levels of $I(k)$. Note that desired levels of $I(k)$ may not be achievable.

5.2 Controller Synthesis Automation Tool

We have developed a Java-based tool, J-DES, as a graphical, multiple-window, platform-independent software package for automating the controller synthesis process in the setting of finite automata representation of the plant. The process is outlined in Figure 1, and shown graphically in Figure 2.

The major advantages of this synthesis tool, compared to TCT (available at http://odin.control.toronto.edu/cgibin-/dlctct.cgi) and Analyzer (available at http://www.engr.uky.edu~kumar/CODE/-java.tar.gz), are the features of interactive

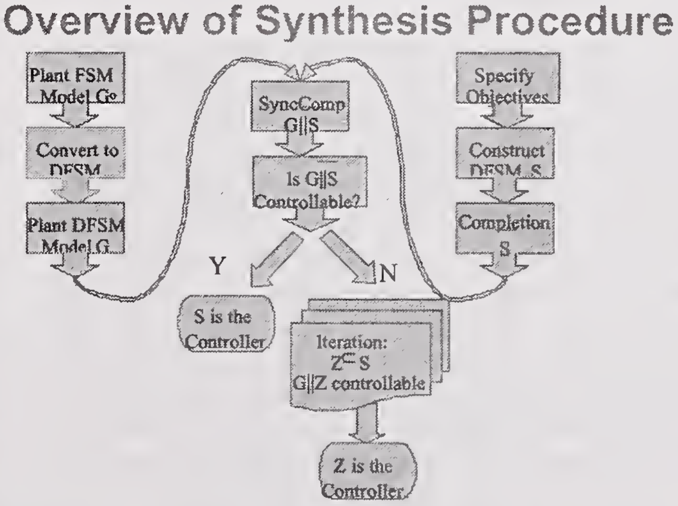


Figure 2. Graphic View of Controller Synthesis

visualization and flexibility. The tool allows the user to create finite automata graphically on a canvas within a window interface by positioning states with single or multi-symbol transitions that always point to the destination state.

The tool allows design for both modular control (i.e., horizontal task decomposition) problems and multi-level hierarchical control (i.e., vertical task decomposition) problems. The detailed design examples using this tool can be found in [Xi 00]. So far for two consecutive levels in the hierarchy, the high-level virtual plant model, which is an abstraction of the low-level closed-loop system in a mealy machine representation, is constructed manually in the J-DES environment. It is this abstraction procedure (i.e., observer computation) that guarantees the hierarchical control consistency.

We have developed an algorithm to extend this tool to optimize a controller for extra sensory intelligence. It is based on the fact that both the plant and controller are based on finite state automata.

Start with a controller, *K*, that controls a plant, $P=(G,a_i)$.

Let $P=\bigcup_{k=0}^{k_{\max}} P'_k$, where P'_k is any k^{th} generation variation of plant *P*, be a population of plants. The 0^{th} generation is defined as the original plant, *P*. Also, let $K=\bigcup_{j=0}^{j_{\max}} K'_j$ where K'_j is any j^{th} generation variation of controller *K*, be a population of controllers. The controller with optimal extrasensory intelligence, $K^m \in K$ is then defined by:

$$m = index\max_j\left(\sum_{i=0}^{k_{\max}} IQ_i(K_j,P)\right)\text{where } \bigcup_j K_j = K.$$

7.0 Results and Open Issues

An experimental testbed implementing a hierarchical controller architecture for a 3-node aircraft command and control network has been implemented under DARPA's JFACC program [Xi 00]. Preliminary results for evaluating extrasensory intelligence for the discrete event controllers designed in this testbed are given in Figure 3.

The following inference may be drawn: $I(k)$ decreases exponentially as *k* increases and may be essentially presumed to be zero beyond $k > 7$.

Further experimentation is required to formulate control mechanisms which possess extrasensory intelligence.

There are fundamental limits to intelligent adaptation in artificial systems. Exploration of these limits, in terms of adaptability of their generating grammars, are open issues. The complexity of the adaptation process is another open issue.

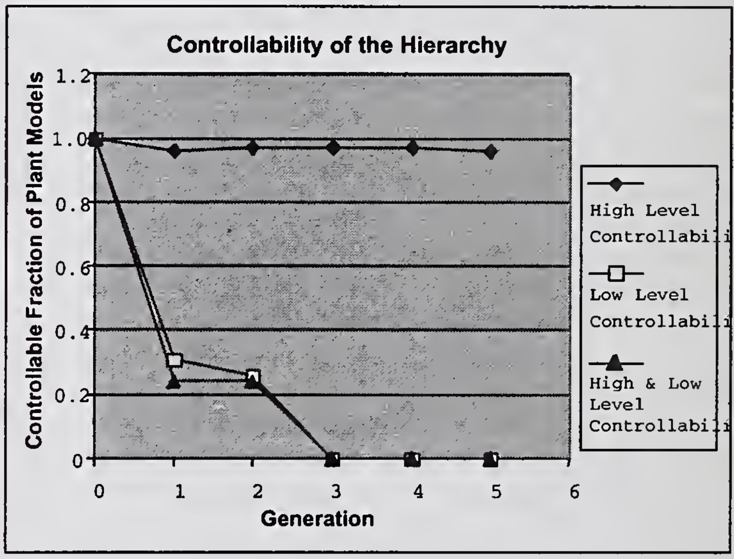


Figure 3. Experimental Results for 3-Node Aircraft Command and Control Network

8.0 Acknowledgment

Some of the concepts of intelligence metrics presented in this paper were initially developed on the JFACC project, sponsored by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-99-1-0547 (JFACC). The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency (DARPA), the Air Force Research Laboratory, or the U.S. Government.

9.0 References

[Albus 93] Albus, J.S., “A reference model Architecture for Intelligent Systems Design,” in *An Introduction to intelligent and Autonomous Control*, pp. 27-56, Kluwer Academic Publishers, 1993.

[Brooks 86] Brooks, R.A., “A Robust Layered Control System for a Mobile Robot, ” *IEEE Transactions on Robotics and Automation*, 2(3): pp. 14-23, 1986.

[Eberbach 99] Eberbach, E., Brooks, R.R., and Phoha, S., “Flexible Optimization and Evolution of Underwater Autonomous Agent, ” *Rough Sets Fuzzy Sets Data Mining and Granular Soft Computer 1999*, Yamaguchi, Japan, Springer Lecture Notes in Artificial Intelligence, Berlin, 1999.

[Friedlander 00] Friedlander, D., Phoha, S., and Ray, A., “Domain Independent Measures of Intelligent Control,” *Performance Metrics for Intelligent Systems Workshop*, to be held in Gaithersburg, MD, August 14-16, 2000.

[Harris 94] Harris, C.J., ed., *Advances in Intelligent Control*, Taylor & Francis, Bristol, PA 1994.

[Hopcroft 79] Hopcroft, J. E. and Ullman, J.D., *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, 1979.

[Levis 93] Levis, A.H., "Modeling and Design of Distributed Intelligence Systems," in *An Introduction to Intelligent and Autonomous Control*, pp. 109-128, Kluwer Academic Publishers, Boston, MA, 1993.

[Meystel 93] Meystel, A., "Autonomous Mobile Robots: Vehicles with Cognitive Control," *Proceedings of the World Scientific*, Singapore, 1991.

[Peluso 96] Peluso, E., "A Hierarchical Structure of Interacting Automata for Modeling Battlefield Dynamics: Controllability and Formal Specification," Ph.D. Dissertation. *Department of Computer Science, The Pennsylvania State University*, 1996.

[Phoha 92] Phoha, S., Sircar, S., Ray, Al., Mayk, I., "Discrete Event Control of Warfare Dynamics, " The Technical Proceedings of the *1992 Symposium on Command and Control Research and the 9th Annual Decision Aids Conference*, Monterey, CA, 8-12 June 1992.

[Phoha 98] Phoha, S., Eberbach, E., Peluso, E., and Kiraly, A., "Coordination of Engineering Design Agents for High Assurance in Complex Dynamic System Design," *Invited paper for Special Track on High Assurance in Intelligent System, 3rd IEEE High Assurance Systems Engineering Symposium*, November 13-14, Washington, DC, 1998.

[Phoha 00] Phoha, S., Gautam, N., Horn, A. "Tactical Intelligence Tools for Distributed Agile Control of Air Operations," *Technical Proceedings of the 2nd IEEE, DARPA-JFACC Symposium on Advances in Enterprise Control (AEC)*, published by IEEE, Minneapolis, MN, July 10-11, 2000.

[Ramadge 87] Ramadge, P.J., Wonham, W.M., "Supervisory Control of a Class of Discrete Event Processes," *SIAM J. Control and Optimization*, Vol. 25, No. 1, January 1987.

[Ray 93] Lee, S., Ray, A., "Performance Management of Multi-Access Communication Networks," *IEEE Journal of Selected Areas in Communications*, Vol. 11, No. 9, pp. 1426-1437, December 1993.

[Ray 95] Garcia, H.E., Ray, A., Edwards, R.M., "Implementation of a Reconfigurable Fault-Tolerant Hybrid Supervisory System," *IEEE Transactions on Control Systems Technology*, Vol. 3, No. 2, pp. 157-170, June 1995.

[Takai 99] Takai, S. "Synthesis of Robust Supervisors for Prefix-Closed Language Specifications," *IEEE Conference on Decision and Control*, pp. 1725-1730, Phoenix, AZ, December 1999.

[Xi 00] Xi, W., Ray, A., Zhang, H., and Phoha, S., "Hierarchical Consistency of Supervisory Command and Control of Aircraft Operations," Technical Proceedings of the *2nd DARPA-JFACC Symposium on Advances in Enterprise Control (AEC)*, published by IEEE, Minneapolis, MN, July 10-11, 2000.

[Zhong 90] Zhong, H. and Wonham, W. Murray, " On the Consistency of Hierarchical Supervision in Discrete-Event Systems," *IEEE Transactions on Automatic Control*, Vol. 35, No. 10, October 1990.

DISTRIBUTED INTERNET-BASED MULTI-AGENT INTELLIGENT INFRASTRUCTURE SYSTEM

Xiaoli Qin[1] **A.E.Aktan**[2] **Kirk Grimmelsman**[3] **F.N. Catbas**[4] **Raymond Barrish**[5] **M. Pervizpour**[6] **E. Kulcu**[7]
Drexel Intelligent Infrastructure And Transportation Safety Institute
Drexel University
3201 Arch Street, Suite 240
Philadelphia, PA 19104

ABSTRACT

The Commodore Barry Bridge (CBB) is a major long-span, cantilever through truss bridge owned by the Delaware River Port Authority (DRPA). To evaluate the performance of this bridge, it is necessary to implement an appropriate health monitoring system, conduct structural analysis, measure the operating and loading environment as well as the critical responses of the structure. The health monitoring system may be used in order to track operational anomalies, deterioration or damage indicators that may impact service or safety reliability. The knowledge space required to accomplish such complex engineering tasks is infinite and uncertain. This engineering domain itself is not well understood. To solve such engineering problems, not only is theoretic knowledge required but also extensive heuristic experience. Organizing and formalizing the theoretical knowledge and heuristic experiences of multidisciplinary human experts is the first major challenge. The building of an intelligent system that can reason and make rational decisions based on induction/deduction of theoretic knowledge and analysis of heuristic experiences is the second major challenge.

This paper presents the writers' progress towards the development of an intelligent infrastructure system that uses integrated technologies of Case-Based Reasoning (CBR) and Rule-Based Reasoning (RBR) to evaluate the performance of the CBB. The system includes a case-base, a rule-base, a CBR agent, a RBR agent and an inter-operational agent. The CBR agent and RBR agent work with both case-base and rule-base. The case-base and rule-base are inter-related through the index schemes. The inter-operational agent evaluates the outputs of the CBR agent and RBR agent to make decisions. This agent can be an alternative human engineer. The CBR methodology is well suited to formulate human experiences and phenomena that would not lend themselves to organization and extraction in terms of rules. In contrast, the theoretical knowledge can be organized using RBR technique. The combination of CBR and RBR technologies offers promise for developing a methodology for solving complex real-life engineering operation problems. The CBR and RBR agents are implemented and wrapped according to CORBA (Common Object Request Broker Architecture) /DCOM (Distributed Component Object Model) standards in order to communicate with each other and an external CORBA server or DCOM server to acquire necessary knowledge.

INTRODUCTION

Case-Based Reasoning (CBR) techniques are a promising for solving many engineering problems. CBR is a subfield of Artificial Intelligence (AI) that is premised on the idea that past problem-solving experiences can be reused and learned from in solving new problems. Rule-Based Reasoning (RBR) techniques are commonly used for developing expert systems in terms of building rules for solving generic or specific problems.

[1] Web: http://www.mcs.drexel.edu/ gxqin; Email: xq22@drexel.edu.
[2] Web: http://www.di3.drexel.edu; Email: Aktan@drexel.edu.
[3] Web: http://www.di3.drexel.edu; Email: grimmeka@drexel.edu.
[4] Web: http://www.di3.drexel.edu; Email: fncatbas@drexel.edu.
[5] Web: http://www.di3.drexel.edu; Email: barrisra@drexel.edu.
[6] Web: http://www.di3.drexel.edu; Email: Mesut.Pervizpour@drexel.edu.
[7] Web: http://www.di3.drexel.edu; Email: Eray.Kulcu@drexel.edu.

This paper discusses the use of combining case-based reasoning and rule-based reasoning techniques to build a multi-reasoning (multi-agent) system to solve a complex domain-specific problem — namely, evaluating bridge performance in civil engineering applications. This paper presents a three-phase approach to building such a system for this domain:

1. *Knowledge Representation for Evaluating Bridge Performance*: building a knowledge-base;
2. *Case-Based Reasoning Engine and Rule-Based Reasoning Engine*: design of the CBR reasoner and intergration of the existing RBR reasoner;
3. *Implementation Issues*: illustrations of how a multi-agent system can be used during the phase of evaluating bridge performance.

Foundation of Case-Based Reasoning and Rule-Based Reasoning Techniques

The *Case-Based Reasoning Cycle* (1) precisely defines a methodology to build a CBR system for a given domain. A case-based reasoning system can be viewed as a model which is a combination of a *case-base* and *knowledge reasoning* process modules. These modules form a *case-based reasoning shell*, also called a *reasoner*. They are the functions used to manipulate the knowledge in the case-base and they act to *process* user inputs, *recall* similar cases, *retrieve* the most similar case, *evaluate and adapt* the retrieved case and update the case memory. The modules interact with the case-base during processing.

Normally, following problems are involved in a CBR system: **knowledge acquisition, knowledge representation, case retrieval, case adaptation** and **the learning mechanism**.

1. **Knowledge acquisition:** How to acquire useful knowledge from application problem domain.
2. **Knowledge representation:** How to use a formal language to represent certain domain knowledge. The knowledge representation theory of case-based reasoning systems primarily concerns how to structure knowledge stored in the case-base to facilitate effective searching, matching, retrieving, adapting and learning. One influential knowledge representation model is the *dynamic memory model* (11). It was developed by Schankand based on his theory, Memory Organization packet (MOP) theory.
3. **Case retrieval:** How to efficiently retrieve from the case-base the case most similar to the current problem. There are two sub-processes involved in case retrieval: one is to retrieve a set of similar cases from case-base, another is to find the most similar case in this set. The first sub-process is accomplished by designing appropriate index scheme for the domain problem. The second task is done using the *Nearest Neighbor Matching Algorithm (NNM)* (7).

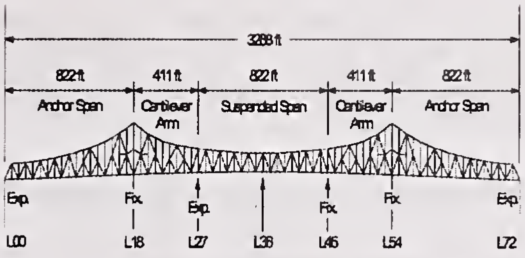


Figure 1: Overview of the Commodore Barry Bridge

Figure 1. Overview of the Commodore Barry Bridge

4. **Case Adaptation Strategies:** After a CBR system retrieves the most similar case from the case-base, it normally needs to perform adaptation on this retrieved case. There are several adaptation strategies which can be used in a CBR system. They are Simple Substitution, Parameter Adjustment and Constraints Satisfaction (7).
5. **Learning Mechanism:** Learning is the last step in the Case-Based Reasoning system. In a CBR system, after a new problem is solved, the case-base is changed by adding the new case into it. By doing that, the system can retain more and more knowledge along with problem-solving augmentation and achieve learning.

For a RBR system, following problems are involved: **knowledge acquisition, knowledge representation, pattern matching definition** and **execution when pattern matching**. The first two problems have the same characteristics as CBR system. For the next two problems, brief explainations are given below:

1. **Pattern matching definition:** How to find patterns which are stored in rulebase. This is accomplished by implementing Rete matching algorithm which is introduced extensively by Charles Forgy's PhD dissertation
2. **Pattern matching definition:** How to excute actions when RBR inference finds applicable patterns. The inference engine loops through all matched rules and fires exhaustively until no more applicable rules in the rulebase.

Domain Problems and Knowledge Representation

Commodore Barry Bridge. The Commodore Barry Bridge (CBB) is owned by the Delaware River Port Authority (DRPA). It links Chester, Pennsylvania with Bridgeport, New Jersey and was opened to traffic in 1974. The bridge is the 3rd longest cantilever truss bridge in the world with a main span of 1,644 feet and a total bridge length of 13,912 feet. Figure 1 shows the principal structural system of the CBB.

Presently, the Commodore Barry Bridge carries more than 6 million vehicles annually, much of it heavy truck traffic seeking to avoid the traffic congestion of the busy Philadelphia metropolitan area (3). The bridge owner wished to objectively evaluate this aging and heavily loaded structure.

The Instrumented Monitoring of the Commodore Barry Bridge. The Drexel Intelligent Infrastructure and Transportation Safety Institute (DIII), working in partnership with the DRPA, has been investigating the application of various health monitoring techniques to the CBB. Health monitoring, in the case of civil infrastructure systems, may be considered as measuring and tracking the operating and loading environment of a structure and corresponding structural responses in order to detect and evaluate operational anomalies and deterioration or damage that may impact service or safety reliability. Designing a monitoring system for a long-span bridge was a major challenge for the DIII researchers. In the past two years, DIII researchers have developed and implemented a health monitoring system for the CBB. This system was designed as a first-cut health monitoring system that would measure global and local responses of the structure in critical members and regions of the bridge. The system takes advantage of in excess of 100 data channels to continuously track the loading environment and numerous structural responses of the bridge (3).

The Structural Identification of the Commodore Barry Bridge. A primary step in implementing a successful global health-monitoring system for a bridge is to accurately conceptualize the structural systems. Long-span bridges typically have numerous complex structural details, boundary, movement and continuity systems that require, at the very minimum, identification and understanding from a conceptual perspective in order to design an appropriate health-monitoring system. These systems, when coupled with transient, non-stationary, nonlinear or unknown load effects and responses, create a monitoring situation that is sufficiently complex to justify a conceptualization effort (3).
Conceptualization of the structural systems is most efficiently accomplished through 3D CAD and solid modeling of the structure, site visits, photographs, and heuristics. The Commodore Barry Bridge consists of sixty-three multi-girder approach spans, eleven deck truss approach spans, and a three span cantilevered through truss. The total length of the bridge from abutment to abutment is 13,912 ft (3).
Selecting the most appropriate bridge members to monitor was another major challenge for the DIII researchers.
The DIII researchers have conducted extensive studies to identify critical bridge members. Correspondingly, sensors are installed at those critical locations to monitor important parameters. Figure 2 shows locations of critical members and their instrumentation.

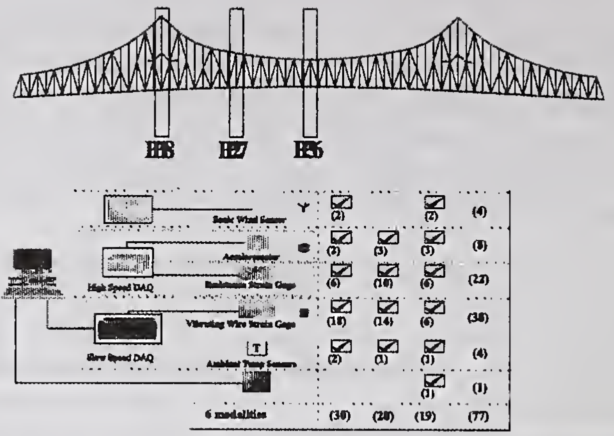


Figure 2. Structural identification And Instrumentation

Why Is An Intelligent Reasoning System Necessary for Health Monitoring? In health monitoring of large structural or infrastructure systems which have the probability of brittle failure modes, engineers are very interested in "intelligent sentries". In the case of the CBB, the researchers have developed sensor systems to monitor the local conditions at the critical regions that are susceptible to fatigue cracking. If these systems sense an incipient cracking, they should inform a human. In this type of effort, a false positive event is very dangerous while a false negative event is totally unacceptable. Therefore, the intelligent agent should be able to follow redundant reasoning and fusion of data from various sensors to rule against false positive while being cognizant of false negative.
The second reason an intelligent agent is needed is for detecting and interpreting the initiation of conditions favorable to deterioration. For example, analysis of internal humidity and electro-chemical characteristics for concrete elements could establish the onset of reinforcing steel corrosion.
Since a health monitor or supervisory control and data acquisition (SCADA) system for a major bridge or a major infrastructure system must utilize many sensors distributed over a large geometric domain, it is impossible for humans to continuously watch for continuously watch for incidents, events and complex phenomena pointing to out-of-ordinary incidents, events and complex phenomena pointing to out-of-ordinary conditions with the structure. The only way a SCADA can become completely effective is if it has self-intelligence to alert human managers when needed.

Multi-Disciplinary Research. The monitor system for CBB has been functioning since 1998, and additional data has been obtained by many controlled tests. Data has been interpreted by the researchers for characterizing the mechanical characteristics and the loading and response environment of the bridge structure in terms of a 3D finite-element model. Researchers continue recording and viewing data from continuous measurements in real-time, and from controlled load tests and ambient vibration tests that are conducted intermittently. The data is used for calibrating the analytical model and validating its reliability for simulating phenomena at the regional and element levels (2), (4), (3), (8).

Research at DIII is conducted in three distinct areas. The first research direction involves investigating, designing and implementing health monitoring systems for civil infrastructure systems. This research is primarily conducted by civil and electrical engineers. The second research area involves structural identification and analysis of instrumented civil infrastructure systems. This requires a team of civil, mechanical, and electrical engineers. The third research area focuses on intergration. This research takes advantage of computer science techniques to fuse distrubuted applications. In addition, knowledge engineering methodology is used to compile, structure and model human knowledge to solve complicated civil infrastructure problems. This research requires the efforts of a computer software engineer. This paper discusses some of DIII's efforts in the third research area.

Knowledge Representation of the CBB . Because of the inherent complexity of the CBB bridge project, the knowledge space in this domain is incomplete and dynamic. It emcompasses civil engineering, electrical engineering and computer science. It is not practical to fully compile and model the knowledge in this project domain. However, acquiring and modeling the primary knowledge for major componts of the project from human engineer is approachable. The major components of CBB project include health monitoring instrumentation and structural analysis. In this paper, a fragment of the knowledge for structural analysis of the CBB and knowledge representation of it is presented. Knowledge acquisition is achieved by specifying only the important features of the problem. Features are only collected if they help solve the specific problem. Other knowledge that is not directly related to solving the problem is discarded. In this approach, a set of important features is predefined for the problem, and knowledge acquisition is done manually by a knowledge engineer. Because of restrictions mentioned above, the system will have some limitations. These limitations will be briefly discussed in the last section

The researchers have conducted extensive research on Case-Based Reasoning (CBR) and Rule-Based Reasoning (RBR) methodologies. Both of these methods provide a very promising way to organize, construct and program human knowledge into a system. This system can contain human experience, theroretical knowledge and respond to the real world based on build-in reasoning mechanisms.

A language called CASL (5) is used to represent knowledge pertaining to CBB project in this case-base. The structure of the case-base is based on Memory Organization Packet (MOP) theory (11).

A language called CLIPS (9) is used to build the rule-base. This language provides three paradigms to organize knowledge which are rule-based, object-oriented and procedure-based.

Case-Based Reasoning and Rule-Based Reasoning Engines

A reasoning engine is software agent which perceives knowledge from the knowledge base, conducts logic inference and reasoning and concludes results. In general, a reasoning engine is used to reason on a specific kind of knowledge base. For example, CLIPS (9) is a tool which provides language used to build a rule-base and a rule-based inference engine is used to reason on the rule-base built by this language.

The case-based reasoning engine is the reasoning system which allows a researcher to use archived cases to solve domain problems. Once domain knowledge has been used to build the case-base, organize memory, build indices, etc., the reasoning engine can execute searches based on the index scheme. The engine can also perform other reasoning processes, including case retrieval, adaptation and system learning.

The rule-based reasoning engine is the reasoning system which reasons on a rule-base. The domain knowledge is compiled, modeled and structured in terms of a series of rules. The rule-based reasoning engine automatically matches facts against patterns and determines which rules are applicable. If they are, the engine performs certain actions specified by knowledge base.

KNOWLEDGE REPRESENTATION FOR CBB BRIDGE PERFORMANCE EVALUATION

Problem Formulation

The CBB project contains two major components. One of them is structural identification and analysis of the bridge. The other is health monitoring of the bridge. In this paper, only a small segment of the domain problem related to the former will be presented as an example. The objective of the structural identification approach is to characterize the as-is structural condition and the loading environment of a bridge through experimental information and analytical modeling. Experimental data acquired from instrumentation is com-

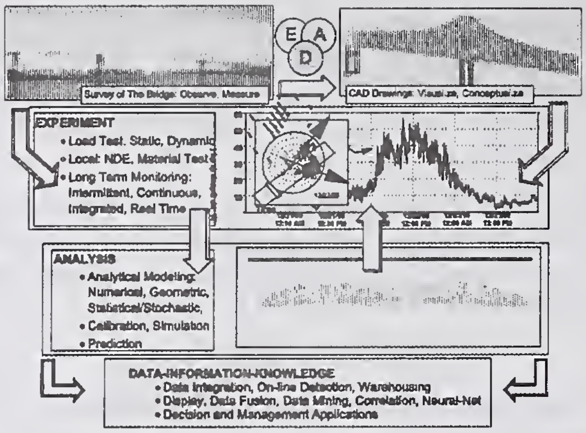

E
A
D
Survey of The Bridge: Observe, Measure
CAD Drawings: Visualize, Conceptualize
EXPERIMENT
• Load Test, Static, Dynamic
• Local: NDE, Material Test
• Long Term Monitoring: Intermittent, Continuous, Integrated, Real Time
ANALYSIS
• Analytical Modeling: Numerical, Geometric, Statistical/Stochastic,
• Calibration, Simulation
• Prediction
DATA-INFORMATION-KNOWLEDGE
• Data Integration, On-line Detection, Warehousing
• Display, Data Fusion, Data Mining, Correlation, Neural-Net
• Decision and Management Applications

Figure 3. Structural And Loading System Identification

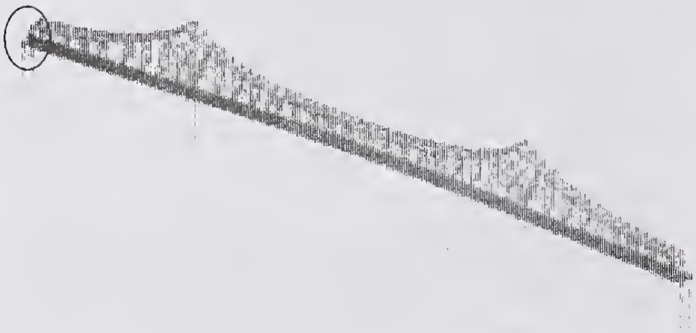


Figure 4. Finite Element Model of the Commodore Barry Bridge

pared to results obtained from an analytical model of the bridge. The results can be used to evaluate the correctness and reliability of the analytical model, to evaluate the performance critical bridge elements, and to issue notifications regarding the safety of structures.

Several 3D Finite Element (FE) models of the CBB were developed to serve as tools for engineering decisions. The analytical models incorporate the contribution of all force resisting elements and mechanisms, particularly those associated with out-of-plane elements, into the analytical model. In this manner, these elements and mechanisms can contribute to the behavior of the model as they do in the actual structure, enabling more realistic and accurate simulations of retrofits, modifications, and loading scenarios (4). The FE models were developed in several stages. First, the structural systems of the bridge were conceptualized by review the design calculations and drawings, shop drawings and site visit. Second, the structure was re-constructed using a 3D CAD model. Finally, the CAD model was transformed into a FE model using a commercial software program. Figure 3 summarizes the development stages and Figure 4 shows the complete 3D FE model of the through truss structure.

DIII researchers conducted a controlled load on the bridge to measure the critical responses. The measured responses provide information necessary to verify analytical models of the structure. The controlled load test on Commodore Barry Bridge consisted of a static load test at pre-identified locations and a crawl speed test. The Commodore Barry Bridge was loaded statically by positioning two large cranes in various configurations. The measured responses included strain measurements for vertical truss members, lower chord truss members, upper chord truss members, floor beams, and for deck stringers near the hangers and midspan regions of the through truss.

The bridge member responses at several locations were measured under a 108 kip crane loading for several loading configurations. The loading configurations were also simulated in the finite element model to obtain analytical responses, which were then compared with the experimental results. The finite element model was found to represent the measured response of the bridge quite well. Examples of model validation and calibration are given in this section along with examples of measurements that illustrate the complexities associated with the response of the bridge. The maximum incremental strain in one hanger due to the prescribed loading condition was measured to be 43.23 microstrain (1.25 ksi). After simulating the same loading condition in the model, the strain value was obtained to be 45.5 microstain. Some formula related above result are given as follows:

$$\sigma = \frac{N}{A}$$

and

$$\varepsilon = \frac{\sigma}{E}$$

where

σ: Stress (ksi)
ε: Strain $(\frac{in}{in})$
A: Cross sectional area of hanger (in^2)
N: Axial load (kips)
E: Young's modulus (ksi)

In the following sections, the hanger analysis of Commodore Bridge will serve as an example to show how to represent the knowledge related to this analysis problem. How to reason knowledge pertaining to it using multi-agent inference engines will also be discussed below.

The CBR Representation Schema

The knowledge pertaining to CBB project can be represented in any kind of representing language. The reason that Case-Based Reasoning Language (5)(CASL) is chosen to represent our domain knowledge is because the knowledge associated problem domain is extremely dynamic and uncertain. Tremendous heuristic experiences are needed to solve practical problems. CASL is a language specially good for model and structure heuristic experiences. The contents of a case-base are described in a file known as a case file, using the language CASL. The reasoner uses this case file to create a case-base in the computer's memory, which can then be accessed and adapted in order to solve problems using Case-Based Reasoning mechanism.

Like any other representing language, CASL has strict syntax, semantics, keywords and operators. The syntax of CASL specifies the grammar rules of organizing knowledge, and the semantics of CASL give the concise interpretation of a sentence written in CASL with correct grammar. CASL defines some basic types in the language: identifiers, strings, numbers and operators, etc..

CASL normally divides a case-base into several modules, each of which has its own syntax features and semantic explanations.

CASL semantics define the meaning of a sentence by specifying the interpretation of the keywords and basic types, and specifying the meanings of operators. In the syntax blocks of CASL, all keywords and literals are given in bold type.

A small example about hanger analysis is provided to show how to use CASL to represent domain knowledge. When a problem is presented, certain conditions are specified. These specifications are the input to the problem solver, or *CBR reasoner*. The CASL structures the knowledge about input problems by defining the primary features of a problem. Every primary feature has a weight associating to it. This weight vaule indicates the importance of this feature.

The brief explanations of primary modules and examples are given below:

1. Introduction. This module defines introductory text which is displayed when the reasoning process (reasoner) is run. The purpose of the text is to help the user understand the contents of the case-base or anything else of note.

2. The Case Definition. The purpose of this block is to define the problem features contained in a case.

In the hanger analysis problem, the most important features are axial forces and bridge type. These features' weight values are set to be 5 (reference weight). The cross sectional area of hanger and Young's modulus etc., are not that important, comparatively speaking. Therefore, their weight values are set to be 0 (reference weight). A sample case definition using CASL is given below:

```
case definition is
field axial-force type is (number) weight is 5;

field bridge-type type is (Long-Span (Suspension, Cable-Stayed, Truss, Arch), Short-Span, Culvert) weight is 5;

field axial-force type is (number) weight is 5;
field cross-section type is (number) weight is 0;
field Youngs-modulus type is number weight is 0;
field Experimental Data type is number weight is 0;
end;
```

3. Index Definition. The purpose of this module is to define which fields are to be used as indices.

This part defines the fields which are used as indices when searching for a matching case. The index scheme defines the methods by which the reasoner should access the case memory. Indices are intended to streamline the matching process. The index features are parts of the new problem specification. For example, we use the features *axial-force* and *bridge-type* as main indices to search the knowledge-base. The sample representation is given below:

```
index definition is
index on axial-force;
index on bridge-type;
```

4. The Adaptation Rule Definition. The purpose of this block is to define rules used to modify a retrieved case from the case-base to make it fit the current problem specifications. The *global repair rule definition* defined in this module allows adaptation rules to be applied on any modified case. The rules defined here are derived from domain knowledge, formulae and constraints.

When the old hanger analysis whose "description of problem definition" part is the most "similar" to the current problem definition is retrieved from the case-base, its solution part must be modified to fit the current problem definition. The reasoner performs adaptations to an old solution according to certain rules defined by domain experts. The **repair rule definition** is block of CASL can be used to define those rules. In the hanger analysis problem, the following rules (strategies) are defined:

(a) Perform simple parameter substitution: substitute parameters of old problem definition into new user input.

(b) Perform old solution adjustment to make it fit substituted user input (current problem) according to domain formulae.

(c) Check global constraints defined in the case-base to guarantee that no conflicts result.

In the sample given in Algorithm 1, the *change_value_1* is an adaptation rule. It tests a certain condition (represented by a formula) first; when the condition is satisfied, the action is fired.

Algorithm 1: Adaptation knowledge representation:

```
repair rule definition is
repair rule change_value_1 is
when
axial-force $\geq$ radial-force
then
evaluate Stress Of Hanger $\sigma$ to $\frac{N}{A}$
evaluate Strain Of Hanger $\varepsilon$ to $\frac{\sigma}{A}$
repair;
end;
end;
```

5. Case Instance Definition. The purpose of this block is to define the structure of a case instance. A case must contain two parts: the problem part and the solution part. The *local repair rule definition* defined in this module allows adaptation rules to be associated with a case. These rules are invoked after the *global* adaptations have run their course.
The past experiences of hanger analysis for applications are stored in the case-base. Representation of these experiences requires the design of certain structures which can represent cases properly. Normally, an experience (case) includes a problem statement part and a solution part. The **case instance is** block of CASL provides a kind of structure and function. This block defines the same structure of problem statement as the **case definition is** block defines.
A sample representation of a case is Algorithm 2:

Algorithm 2: Case Instance Representation:

```
case instance Hanger Analysis is
bridge-type = Truss;
axial-load = N;
cross-section = A;
Young-Modulus = E;
Experimenta Data = D;
solution is
Stress = S1;
Strain = S2;
permissable capacity= C;
local repair rule definition is
repair rule rule_1 is
when
bridge-type $\neq$ Truss
then
pr 'Abandon your selection ! ';
pr 'This case can not be repaired to let you use!';
reselect;
repair;
end;
end;
```

The RBR Representation Schema

CLIPS (C Language Integrated Production System) is an expert system tool developed by the Software Technology Branch (STB), NASA/Johnson Space Center (9) . It is designed to facilitate the development of software to model human knowledge or expertise. There are three ways to represent knowledge using CLIPS in a rulebase:

(a) **Rule-Based Knowledge Representation:** In this paradigm, knowledge is represented as a series rules. Rules are used to represent heuristics which specify a set of actions to be performed for a given situation. A rule is composed of a *if* part and *then* part. The *if* part is a set of patterns which which specify the facts which cause the rule to be applicable. The process of matching facts to patterns is called pattern matching (9). The built-in inference engine matches facts against patterns and determines which rules are applicable.

(b) **Object-Oriented Knowledge Representation:** In this paradigm, knowledge is represented as a series modular components which inherit object-oriented mechanism. For example, this mechanism makes hierarchy knowledge models possible.

(c) **Procedural Knowledge representation:** In this paradigm, knowledge is represented in terms of procedural style like conventional language C, C++ and Pasal etc. This capability is extremely useful when knowledge can not be represented using rules or object-oriented mechanism.

Example of Rule-Based Knowledge Representation. The following example shows how a rule-based representation is used for the CBB hanger analysis. The example shows that when the RBR inference engine finds experimental data from instrumented hanger that is much larger than analytical data from FE model, it fires and generates warning signal.

Algorithm 3: Rule-based representation by CLIPS:

```
(defrule Warning-Signal)
(experimental-data-isData)
(analysis-data-isA-Data)
assert(Something wrong on the CBB bridge!)
```

MULTI-AGENT REASONING SYSTEM

System Overview

The *case-based reasoning engine*, also called *reasoner*, takes problem specifications and a case-base file as its inputs, performs reasoning about the problem, and returns an answer to the user automatically. The reasoning engine of a case-based system consists of four process modules; each of those modules performs certain functions. The modules interact with the case-base and form a reasoning cycle. The first module, *Retrieved case*, takes the current problem specifications as input and outputs a retrieved case. The second module, *Solved case*, decides whether a retrieved case needs to be adapted. This module either returns to the user a solution without further modification or passes a solution to the next module which will perform adaptation on the case. The third module, *Repaired case*, performs this adaptation and returns an adapted case to the next module. The fourth module, *Learned case*, decides whether this new resolved case needs to be stored in the case-base. The kernel of CBR engine used in the problem domain was developed by Center for Intelligent System. University of Wales (5). The primary author has developed a wrapper for this engine, added extra features for this engine and has added extra features including a Graphical User Interface (10).

The *rule-based reasoning engine* which is part of the CLIPS system, also called inference engine, was developed by NASA's Johnson Space Center (9). The CLIPS was a *forward chaining rule* language based on the *Rete pattern matching* algorithm. The inference engine was implemented as different modules. Every module has different functions and purposes. Detailed information about these modules is provided at the CLIPS website (9). Only basic ideas of this inference engine are introduced here. When the inference engine is invoked (perceives input from outside world), it automatically looks at the rulebase and matches facts against patterns which are defined by the knowledge engineer. It then determines which rules are applicable. It selects a rule and then the actions of the selected rule are executed. The inference engine then selects another rule and executes its actions. This process continues until no applicable rules remain (9).

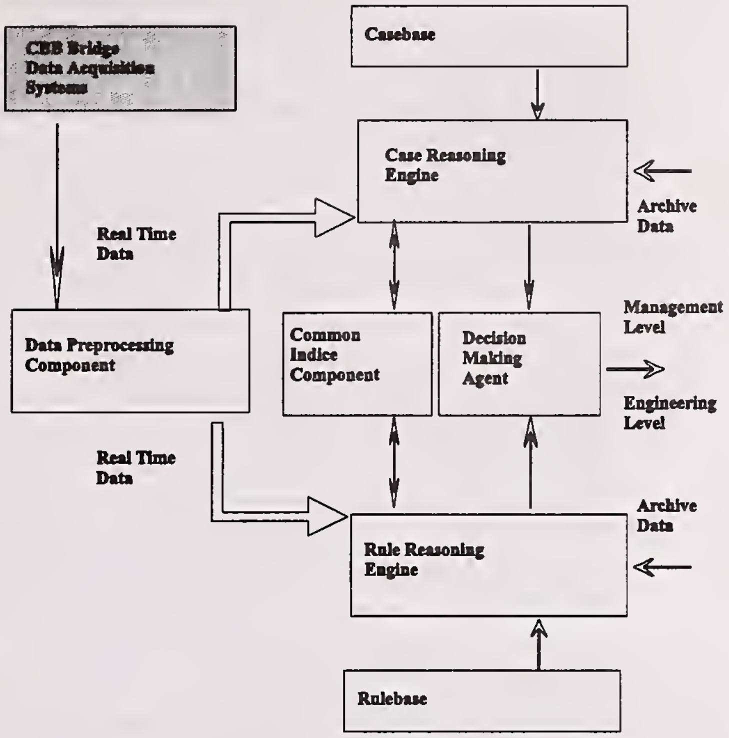


Figure 5. Multi-Agent Reasoning System Architecture

The following sections present how the writers will integrate and implement these modules and the RBR inference engine. Only a detailed introduction of CBR reasoning engine is presented here.

System Architechures

Main System Architecture. Figure 5 shows the architechture of multi-agent reasoning system.

There are seven modules in the system. Each module performs specific tasks and functions.

(a) **CBB Bridge Data Acquisition Module:** This module collects data from instrumented sensors on the CBB bridge and performs buffering and raw data storage functions. The module consists of several different data acquisition hardware and software systems acquired from a variety of vendors. Some of the vendors provide built-in functions which can be integrated with commercial data processing software. This makes communication between the *CBB Bridge Data Acquisition Module* and the *Data Preprocessing Component* discussed below possible. For example, the OPTIM Electronics data acquisition system (6) provides

OLE interface that enables the system to communicate with the commercial data processing software LabView which also supports the OLE standard.

(b) **Data Preprocessing Component:** This component performs data preprocessing. It takes the *CBB Bridge Data Acquisition Module* as input, checks data quality, eliminates electrical signal errors and conducts preliminary data analysis and other related tasks. This module's kernel could be some data processing software like LabView. The module outputs data either immediately to a display through some interface like web browser or automatically to a database. For example, the former can be implemented through LabView that can read data from OPTIM and send it to a web browser for direct and immediate display to the user. The latter can be implemented by writing a customized program which reads data from LabView's buffer or from its data storage disk. The program then routes the data to an archived database.

(c) **Case-Based Reasoning Engine:** This module performs CBR reasoning. It perceives knowledge from casebase, extracts data from both the *Data Preprocessing Component* , an archived database and the *Common Indice Component.* It performs reasoning based on all the information mentioned above and returns reasoning results to the *Decision Making Agent.*

(d) **Rule-Based Inference Engine:** This module performs RBR reasoning. It perceives knowledge from the rulebase, extracts data from both the *Data Preprocessing Component* , an archived database and *Common Indice Component.* It performs reasoning based on all the information mentioned above and returns reasoning results to *Decision Making Agent* which will be introduced below. This module will be presented in more detail later. .

(e) **Common Indice Component:** This component serves as hub to connect CBR system and RBR system together. It takes the user's problem as input, checks important features of domain problem and outputs these features to the CBR engine and the RBR engine. It triggers both engines through industry standard protocols such as COM(DCOM) which allows distributed applications to communicate with each other. Since the source code for the CBR and RBR engines are publicly available and they were created using C language, it is not difficult to program a wrapper with a standard interface for both engines.

(f) **Decision Making Component:** This component collects the output from the CBR reasoning engine and the RBR inference engine. The component can be a software entity or a human entity. The entity judges the output from both engines and conducts reasonable actions. For example, a software entity can issue warnings based on its judge from outputs of inference engines to warn the human manager that some members of the bridge need to be reinforced or that the bridge should be closed due to some catastrophic event.

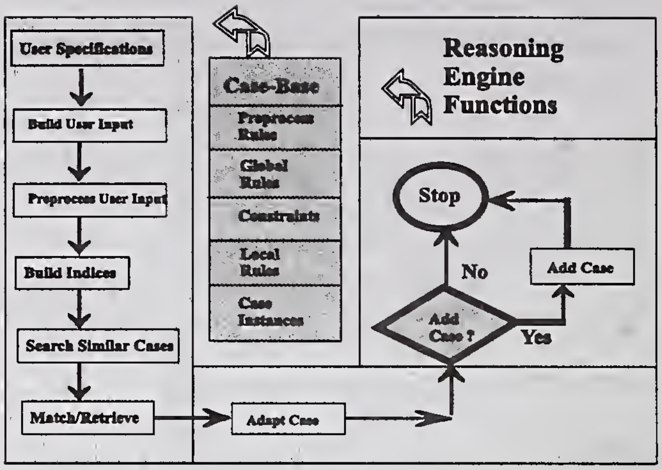


Figure 6. The primary functions of a CBR Reasoning engine

Reasong Process of CBR Engine .

The flow-chart in Figure 6 shows the main algorithm behind the implementation of CBR reasoning engine. The two hollow arrows in the figure illustrate that the reasoning engine must interact with the case-base.

The flow-chart shows that the requirements of a module can be broken into pieces or procedures called by the main function. It also shows that a CBR engine forms a reasoning loop. This reasoning loop begins with the procedure *User Specification* and ends with the procedure *Add Case.* Primary procedures used in the main algorithm are discussed below separately.

Building the Index

The performance of a CBR system is determined by the CBR reasoning engine whose efficiency is in turn determined by the design of the *index scheme* and the *case-base memory organization.* The index scheme design includes how to specify index features and how to build them in computer memory. The index features are set by domain experts and are represented by the block **index definition is** of CASL. The procedure *Build Indices* takes the representations of index features as input and uses these to build the index scheme. A *linked-list* data structure was chosen to hold the index feature input. The procedure *Build Indices* places all the index features into the linked-list, and at the same time, builds the case-base memory organization. Figure 7 illustrates these ideas.

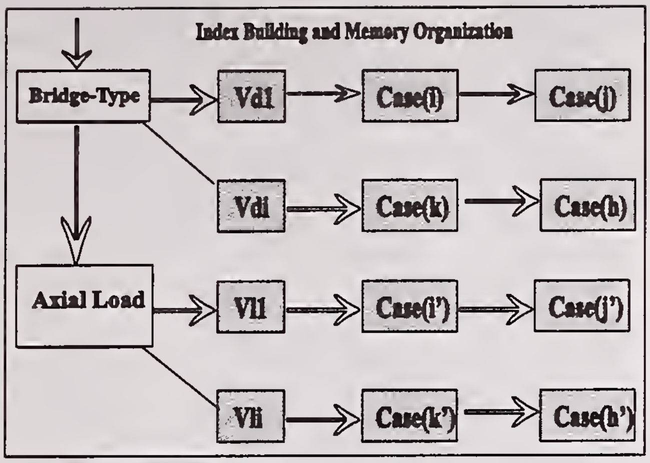


Figure 7. The index building and case-base memory organization

In this CBR system for the hanger analysis problem, two features have been specified as index features: *bridge-type* and *axial-load*. Each index feature is a node of the linked-list and the data type for the nodes is the **struct** type in C. The fields of the **struct** are used to hold attributes of the index features. Figure 7 shows this data structure for the index features and case-base memory. The procedure *Build Index* first links the index features *shaft diameter* and *load direction*. It then checks every attribute of the index features. For each attribute, *Build Index* searches for all the cases with the same attribute value in the case-base file and links all of these together.

Case Matching, Ranking and Retrieving

The purpose of building an index scheme is to speed up *searching*. Here, *searching* means to find a set of cases from the case-base which are similar to the current input case. However, the goal is to find the case that has the maximum similarity to the input case. Thus, a mechanism to rank the similarity of cases is needed. In this section, the procedure necessary to accomplish two goals (finding a similar case set and finding the most similar case in this set) is discussed.

First, a mathematical model is presented to demonstrate how to find a set of similar cases in the case-base. What are *similar cases*? **Given an input case with certain index features and their attributes, similar cases are those cases whose index features and attributes are exactly the same as the corresponding input case's.** Figure 8 shows these ideas.

The top portion of the Figure 8 illustrates the mathematical model for finding similar cases. The left and right circles represent attributes $F(A)$ and $F(B)$ of index features A and B of an input case respectively. The $C(n)$ represents a case n. If the left circle includes $C(b), C(d), C(h)$ and $C(a)$, which are the cases with attribute $F(A)$ of feature A, and the right circle includes $C(i), C(j), C(a)$ and $C(h)$, which are the cases with attribute $F(B)$ of feature B, then their intersection contains cases $C(a)$ and $C(h)$, which have both attribute $F(A)$ and $F(B)$. This can be represented in set theory as:

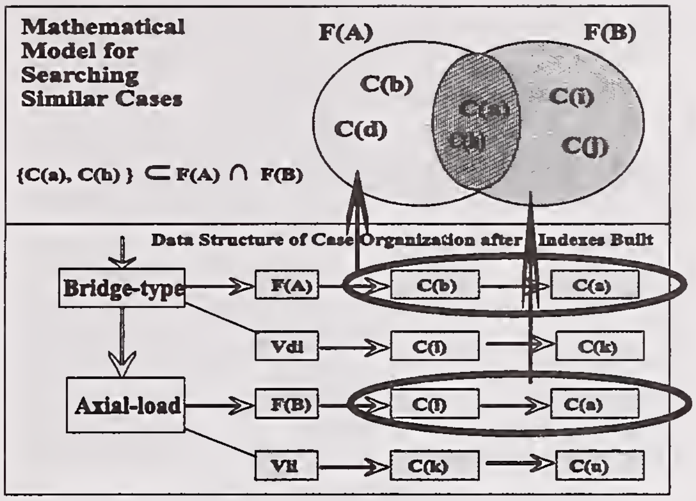

Mathematical Model for Searching Similar Cases
F(A)
F(B)
C(b)
C(d)
C(a)
C(h)
C(i)
C(j)
{C(a), C(h) } ⊂ F(A) ∩ F(B)
Data Structure of Case Organization after Indexes Built
Bridge-type
F(A)
C(b)
C(a)
Vdi
C(l)
C(k)
Axial-load
F(B)
C(l)
C(a)
Vli
C(k)
C(n)

Figure 8. The mathematical model and an example for searching similar cases

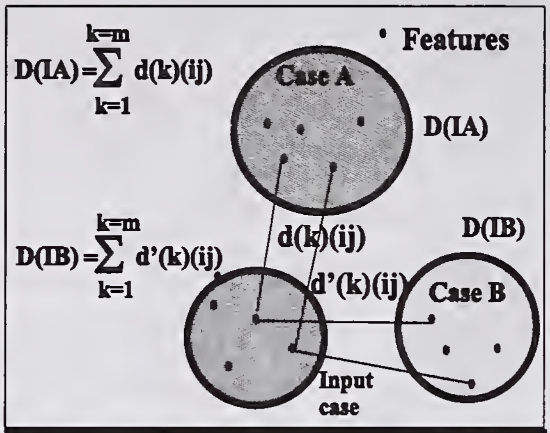
$$\{C(a), C(h)\} \subset F(A) \cap F(B)$$

The bottom portion of Figure 8 provides a corresponding example to illustrates how this process occurs in the case-base.

After all similar cases are found, a mechanism to find the most similar case in this set is needed. In the system, the **Nearest Neighbor Matching algorithm (NNM)** (7). Figure 9 illustrates how this algorithm works in the CBR system for the hanger analysis. To simplify the discussion, it is assumed that all of the component loads applied to the hangers are at in the same direction.

The basic idea of the NNM algorithm is to compare the attribute value of each feature for each case in the set of similar cases to every corresponding feature's attribute of the input case, calculate the comparison values and then sum them for each case to get a total comparison value.

In the upper portion of Figure 9, the circles represent cases, the dots represent attribute values of features, index i represents the input case, and index j represents cases in the set of similar cases. The index k represents the features in a case. The case A and case B in the figure are the cases from the similar cases' set. The function $d(k)(ij)$ represents the attribute's comparison value of one of the features (feature k) between the input case and case A, which is equivalent to


$D(IA) = \sum_{k=1}^{k=m} d(k)(ij)$
• Features
Case A
D(IA)
$D(IB) = \sum_{k=1}^{k=m} d'(k)(ij)$
d(k)(ij)
D(IB)
d'(k)(ij)
Case B
Input case

Relevance Matrix

F(k)(R)i F(k)(I)j	V(k)(R)1 • • •	V(k)(R)n

V(k)(I)1
•
:
	d(k)(11)=
	1 * W(11)
		d(k)(1n)=
		Sim(F(R)i, F(I)j)
		* W(1n)

V(k)(I)n	d(k)(n1)=
	Sim(F(R)i, F(I)j)
	* W(n1)
		d(k)(nn)=
		1 * W(nn)

Sim(F(k)(R)i, F(k)(I)j) = 0, ¾, (1- Δd / Δr)

Δd/Δ r: difference of feature value / difference range value between input and retrieved case feature

Figure 9. The Nearest Neighbor Matching algorithm

the following formula (7):

$$W(ij) * Sim(F(k)(R)i, F(k)(I)j)$$

where:
k: a feature of a case.
$W(ij)$: the weight of a feature, defined in the case-base file.
$Sim(F(k)(R)i, F(k)(I)j)$: the degree of similarity between one of the features in the input case and the corresponding feature in a case from the similar case set.
The total attributes' comparison value for a case is D(k)(IA), which is equal to the numeric function

$$\sum_{k=1}^{n} W(ij) * Sim(F(k)(R)i, F(k)(I)j)$$

After finishing all calculations, the NNM algorithm selects the case which has the highest value of $D(k)(ij)$ to be the most similar case.
The key component of the NNM algorithm is the calculation of an attribute's comparison value for a feature between a similar case and the input case. A matrix called the *relevance matrix*, shown in the lower part of Figure 9, is used to explain how to calculate every feature's attribute comparison value. In the matrix, $F(k)(R)i$ means "the feature k of a case from the similar case set which has possible attribute i, where the range of i can be from 1 to some finite number". $F(k)(I)j$ has a similar meaning except in reference to the input case. So, the first row of the matrix represents all the possible attributes of feature k of a similar case, and the first column represents all the possible attributes of feature k of the input case. The intersection of row and column is the comparison value of the feature k. The $W(ij)$ is the weight of a feature in a similar case. The degree of similarity $Sim(F(k)(R)i, F(k)(I)j)$ has three possible values. First, if two features match exactly, the degree of similarity equals 1. Second, if two *abstract symbols* are similar, its value is $\frac{3}{4}$. Third, if two *numbers* are similar (i.e., both fall within the range defined in the *modification* block), then a value is calculated which reflects how close they are in proportion to the range. Then, the $Sim(F(k)(R)i, F(k)(I)j)$ can be calculated by:

$$1 - \frac{\Delta d}{\Delta r}$$

where: Δd is the difference of the feature values between the input case and the retrieved case and Δr is the difference range value. For example, if the attribute value of feature *axial load* for the input case is 64 kips, and the corresponding value for a similar case is 84 kips Newtons, then Δd = 84-64=20. If the definition for the range of similarity is from 44 to 94, then Δr = 94-44=50. *Similarity between 64(input) and 84(a similar case)* $= 1 - \frac{20}{50} = 0.6$
Algorithm 4 defines the functions needed to find similar cases and the most similar case as mentioned above. The procedure *Index_List_Searching( )* performs searching on the linked-list of index features. The procedure *Case_List_Searching( )* searches out cases whose attribute value for certain features is the same as the input case's. The procedure *Computing_Weight_Cases( )* performs calculates the weight of a retrieved case and returns this value. The procedure *Evaluating_Similar_Cases( )* ranks a case with a weight. The procedure *Retrieving_Heaviest_Case( )* retrieves the case with the highest rank and returns this.

Adaptation of Cases

It is rare for a retrieved case to be exactly the same as the newly defined problem. Most of the time the retrieved case is only a similar situation, and so problem definitions and corresponding solutions must be modified so that the modified case fully fits the current situation and its solution fully

Algorithm 4: Case matching, ranking and retrieving:
Input: User's input problem specification.
Output: The retrieved case with highest weight.

```
MATCHING_RANKING_RETRIEVING(UserInput)
begin
while true
    do
    Index_List_Searching( );
    Case_List_Searching( );
    Computing_Weight_Cases( );
    if Case_Matching_Exact = True;
        return Retrieving_Case();
    else
        Evaluating_Similar_Cases( );
        Retrieving_Heaviest_Case( );
        end
```

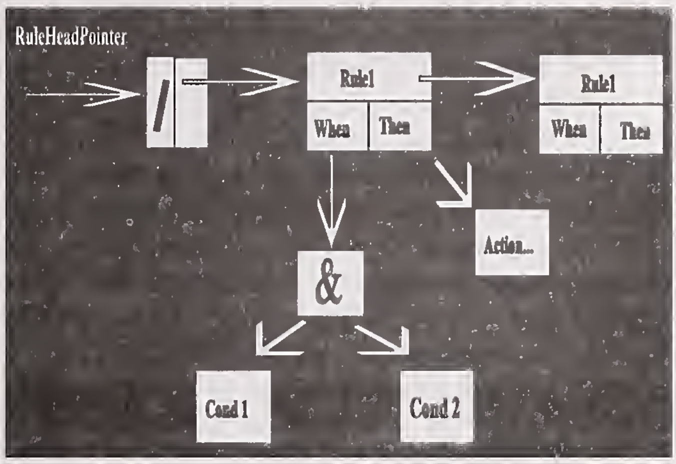


Figure 10. The data structure of global and local rules

Algorithm 5: Algorithm for case adaptation:
Input: Retrieved case.
Output: The modified case.

```
CASE_ADAPTATION(RetrievedCase)
begin
while true
    do
    if Global_Rules = True;
        Finding_Global_Rule_Headpointer( );
        Searching_Global_Rules( );
        Apply_Modifying_Retrieved_Case( );
        Parametric_Adaptation( );
        Constraints_Adaptation( );
        Evaluating_Solutions( );
        return Modified_Satisfied_Case;
        end
```

satisfies the current problem requirements. This procedure as a whole is called the case adaptation (repair) process. A series of rules are defined for adapting cases. These rules are provided by domain experts or domain axioms and are applied to each case whenever it is necessary.

Adaptation rules are divided into *global rules* and *local rules* The reasoner uses *global rules* to examine the problem fields and solution fields of the retrieved case. These rules are also used to adapt the parameters of the retrieved case and check constraint satisfaction conditions which are specified by the knowledge-base. If there are any constraint conflicts, the repair rules provide a new problem-solving proposal. Otherwise, they adapt the solution of the retrieved case to the new problem. Sample adaptation rules for global repair are described in **Algorithm 1**.

Figure 10 shows that a linked-list data structure is used to store these adaptation rules. In the figure, every node has two fields: one stores the condition of a rule, the other stores the action. The procedure given in **Algorithm 5** scans the rule list repeatedly as it performs adaptation on a retrieved case; if the condition part is true, it executes the corresponding actions on the case.

FUTURE WORK

This research touched upon both AI/CBR/RBR and bridge engineering domains. The system discussed can also serve as a template for other engineering domains. The following areas are envisioned for future research.

(a) **Implementation issues:** Several challenges must be overcome in order to implement a multi-agent system. The first challenge relates to knowledge engineering problems which are discussed below. Another challenge arises from the fact that the system is composed of different modules, components and applications which are eventually distributed on different computers and on a network. Developing efficient interfaces and a wrapper to permit them to effectively communicate with each other is a major hurdle.

(b) **Knowledge engineering issues:** Because of the limitations of the CASL and CLIPS used to build our system, there are still many limitations in expressing problem solving intent. The case collection process is quite complicated and inefficient, and case-base maintenance is very unstructured. This makes debugging the case-base very difficult. The rule scope defined for the project domain is extensive because of the complexity of the CBB project. Improved methodologies for case collection, rule collection and better protocols to maintain the case-base and rule-base are needed.

(c) **Knowledge acquisition issues:** The authors built attribute (features) pairs during the initial design to allow the user to interactively input knowledge. If the system is expanded (especially cross-domain), it would be very difficult to enumerate all the features during design time to cover any and all possible problem specifications. Therefore, the development of an autonomous knowledge acquisition system is a future challenge.

(d) **Indexing issues:** The authors built a fixed feature-based index scheme during the initial design to speed up searching. However, as stated above, if the system is expanded, it would be impossible to optimize this choice of index features. as the system is utilized, many additional features may become important primary design factors. Since these features are not initially coded into the case-base or rule-base, the system will fail to find cases or match rules which have these important features. Developing a dynamic index scheme that will address this situation is an additional research need.

(e) **Graphical reasoning issues:** In the Commodore Barry Bridge project domain, many problems are solved by heuristic experience. Such experience is routinely relied on for interpreting processed images or graphics; therefore, a graphical inference capability becomes necessary. How to combine textual reasoning procedures with graphical reasoning procedures is another very important issue.

CONCLUSIONS

This paper discussed a system that uses Case-Based Reasoning and Rule-Based Reasoning as both a cognitive model and problem solving methodology to deal with a bridge engineering problem for civil engineering applications. The authors believe that this work will produce several insights into how AI, CBR and RBR techniques can be better applied to more realistic engineering problems:

(a) **Knowledge Capture:** Because the knowledge space for the Commodore Barry Bridge domain is extremely incomplete and dynamic, it is difficult to strictly rely on formalizing general, *a priori*, rules to help engineers to solve problems or automate the problem solving process. But using CBR techniques, the extensive experience of many experts can be stored in a case library. In contrast, rule-based techniques can compensate for the shortcomings of case-based techniques. Through exploration of a useful rule-based tool like CLIPS, the knowledge can be modeled using object-oriented mechanism.

(b) **Adaptability:** CBR techniques can integrate knowledge acquisition, reasoning mechanisms, knowledge storage and learning in one platform. Therefore, a system using CBR techniques can possibly grow and expand to encompass a wider variety of assemblies without changing the fundamental system structure.

(c) **Augmenting Intelligence:** The proposed system, rather than being completely autonomous, interacts with the user to obtain knowledge. It provides the flexibility to draw conclusions either from the system itself automatically or by allowing the human engineer to decide which actions he/she should take.

(d) **Human-Guided Search:** The system also provides the flexibility to allow the engineer to loosen index constraints to continue reasoning when an exact search fails. In this manner, the engineer has the most opportunities to obtain a solution that is useful for his/her current problem.

Acknowledgements. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of the supporting government and corporate organizations. The authors gratefully acknowledge research support from the following individuals: Mr. Robert Box (DRPA), Drs. Steven Chase and Ghasemi (FHWA), and Drs. S.C. Liu and Ken Chong (NSF).

REFERENCES

[1] Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *Artificial Intelligence Communications*, 7:39–59, 1994.

[2] A. Emin Aktan, Kirk A. Grimmelsman, Raymond A. Barrish, and F.N. Catbas. Structural identification of a long span bridge. *Proceedings of the Fifth International Bridge Engineering Conference*, 1, 2000.

[3] Raymond A. Barrish, Kirk A. Grimmelsman, and A. Emin Aktan. Instrumented monitoring of the commodore barry bridge. *Proceedings of SPIE Fifth International Symposium on Nondestructive and Health Monitoring of Aging Infrastructure*, 2000.

[4] F.N. Catbas, Kirk A. Grimmelsman, and A. Emin Aktan. Structural identification of commodore barry bridge. *Proceedings of SPIE Fifth International Symposium on Nondestructive and Health Monitoring of Aging Infrastructure*, 2000.

[5] Center for Intelligent System. University of Wales. http://www.aber.ac.uk/ dcswww/Research/arg/cbrprojects/getting_caspian.html. Caspian.

[6] OPTIM Electronics http://www.optimelectronics.com, 2000.

[7] Janet Kolodner. *Case-based Reasoning*. Morgan Kaufmann Publishers, Inc., San mateo, CA 94403, 1993.

[8] Eray Kulcu, Xiaoli Qin, Raymond A. Barrish, and A. Emin Aktan. Information technology and data management issues for health monitoring of commodore barry bridge. *Proceedings of SPIE Fifth International Symposium on Nondestructive and Health Monitoring of Aging Infrastructure*, 2000.

[9] NASA. Clips. http://www.ghg.net/clips/clips.html, 1999.

[10] Xiaoli Qin and William C. Regli. Applying case-based reasoning to mechanical bearing design. *2000 ASME International Design Engineering Technical Conferences and the Computers and Information in Engineering Conference*.

[11] C.K. Riesbeck and R.S. Schank. *Inside case-based reasoning*. Erlbaum, Northvale, NJ, 1989.

Stigmergy – An Intelligence Metric for Emergent Distributed Behaviors

Richard R. Brooks
Applied Research Laboratory
The Pennsylvania State University
P.O. Box 30
State College, PA 16803-0030
Email: rrb@acm.org

Abstract

Individual autonomous components can be constructed using simple behaviors based entirely on locally available information. Simple components aggregate to form complex systems with complex behaviors. Artificial life research has proposed guidelines for constructing colonies of autonomous systems. Simulations mimicking biological systems show these guidelines adequately explain the behavior of many insect species. The complexity of aggregated behavior often depends on *stigmergy*. Stigmergy occurs when behaviors by individuals modify the environment while being regulated by the environment's state. Stigmergy has generally been studied for the *forward problem*: predicting the consequences of local behaviors. It is also applicable to the *backward problem*: synthesizing local behaviors to fulfill a global need. The concept provides an objective measure of intelligence for natural and synthetic systems. A system's intelligence is measured by its amount of effective stigmergy. It not only adapts to a changing environment, but also modifies the environment to suit the system's needs and goals.

Keywords: *intelligence metrics, artificial life, stigmergy, distributed intelligence*

1. Introduction

"Self-centered – someone who does not think about me." - Coluche (French Comedian
)

Egotistically, most people consider another person intelligent when the other person agrees with them. The Turing test is an egregious example of this tendency. A system is intelligent, when its behavior resembles human behavior. While flattering, this measure is not very objective. Objectively, intelligence is a combination of many attributes. These include the ability to:

- Achieve goals
- Compete with others
- Cooperate with others
- Develop new unexpected behaviors
- Adapt to a changing environment

These attributes are necessary but not sufficient for describing intelligence. A truly intelligent system should also interact with its environment, modifying the environment to its advantage. This ability is based on what is called *stigmergy* by Grassé [16].

Grassé coined the term stigmergy while studying highly evolved societies of cooperating individuals. The societies shared the following characteristics:

- Construction of climate controlled communal housing
- Individuals altruistically sacrifice themselves for the common good
- Equitable distribution of work among their members
- Division of tasks among castes of specialized workers
- Domestication of other species
- Creation of logistic networks to support cities and war campaigns

These societies belong to the most universally successful species on earth, controlling most of the air and ground space. They are distinguished by having six legs.

A collective view of intelligence is not limited to the behaviors of insect societies. Cellular Automata research shows how networks of extremely simple automata collectively emulate general computation engines, such as Turing and von Neumann machines [27]. Connectionist methods in artificial intelligence create complex behaviors in a network of extremely simple computation engines [10]. Minsky's *Society of Mind* describes human behavior emerging from interactions among multiple simpler individual entities [19].

Bonabeau's work [3, 4, 5] provides a starting point for an objective definition and measure of intelligence. An individual, or society of individuals, is intelligent when it exhibits a significant degree of stigmergy. It not only adapts to its environment, it interacts with the environment, forcing the environment to adapt to its needs and goals. This interaction is not purely deterministic but results in new behaviors that advance the system towards its goal.

The rest of this paper is organized as follows: Section 2 discusses relevant aspects of cellular automata, on appropriate formalism for studying interactions of distributed systems. Relevant studies of insect colonies are provided in section 3, along with the original concept of stigmergy. Section 4 discusses applications of pheromones and stigmergy to synthetic systems. Some applications reproduce lifelike behaviors. Other applications create synthetic environments using stigmergy-like control mechanisms. Section 5 describes a distributed system where simple behaviors of local systems combine to produce complex adaptive behavior for the network. A possible stigmergy scale is further discussed in section 6, which concludes the paper.

2. Cellular Automata

A cellular automata (CA) is a synchronously interacting set of elements (network nodes) defined as a synchronous network of abstract machines [1]. A CA is defined by:

- *d* the dimension of the automata
- *r* the radius of an element of the automata
- *d* the transition rule of the automata
- *s* the set of states of an element of the automata

An element's (node's) behavior is a function of its internal state and those of neighboring nodes as defined by *d*. The simplest instance of a CA is uniform has a dimension of 1, a radius of 1, and a binary set of statcs. In this simplest case for each individual cell there are a total of 2^3 possible configurations of a node's neighborhood at any timc step. Each configuration can be expressed as an integer *v*:

$$v = \sum_{i=1}^{1} j_i * 2^{i+1} \qquad (1)$$

where: *i* is the relative position of the cell in the neighborhood (left=-1,current position =0, right=1), and j_i is the binary value of the state of cell *i*. Each transition rule can therefore be expressed as a single integer *r*:

$$r = \sum_{v=1}^{8} j_v * 2^v \qquad (2)$$

where j_v is the binary state value for the cell at the next time step if the current configuration is *v*. This is the most widely studied type of CA. It is a very simple many-to-one mapping for each individual cell. The aggregated behaviors can be quite complex [11]. Wolfram [27] has created four qualitative complexity classes of CA's:

- *Stable* - Evolving into a homogeneous state.
- *Repetitive* - Evolving into a set of stable or periodic structures.
- *Chaotic* - Evolving into a chaotic pattern.
- *Interesting* - Evolving into complex localized structures.

Two further results show the computational abilities of the CA. Simple CA's can be constructed that reproduce themselves. This was one of the initial concepts von Neumann had in mind when he originated the CA model [11]. CA networks of sufficient size are capable of simulating general computations [17]. Networks containing interactions of extremely simple automata are therefore capable of producing arbitrarily complex aggregated system behavior.

This is related to the ability of neural networks to produce complex behaviors through network interactions among simple threshold devices. Feed-forward and competition networks can infer complex piecewise linear classification functions from a set of examples [10, 22]. These abstractions support the concept that intelligence is a property of aggregated system interactions, rather than individual components. It is worth noting that most connectionist approaches rely on randomly choosing initial conditions in the network.

3. Insect behaviors

Artificial life researchers seek new approaches to intelligence, coordination, and self-organization among distributed autonomous systems in insect colony behaviors [24, 21]. Self-organization is very important in living systems. The basic ingredients of self-organization are [3]:

- Positive feedback - includes recruitment and reinforcement of behaviors.
- Negative feedback - counterbalances positive feedback to stabilize the system.
- Amplification of fluctuations - randomness and fluctuations are crucial to system adaptation.
- Multiple interactions - simple behaviors at the micro level aggregate into intelligent adaptations at the macro level.

In addition, arthropods have a number of broadcast signals such as alarms [25]. These primitives are biologically inspired and the basis of many complex animal behaviors such as swarming, flocking, etc.

Insect colonies use pheromoncs to provide positive and negative feedback signals with thesc characteristics [14]. Pheromones are natural chemicals secreted by individual animals, and received by other individuals using the sense of smell. They influence the behavior and development of thc reccivers. Pheromone interactions have been used to model food collection, nest building, task allocation, and war in insect societies [3]. Computer simulations based on these explanations have produced colony bchaviors similar to those found in nature [5].

Stigmergy is indirect communication between one or more agents through the environment using pheromone interactions [16]. An individual interacts with its environment depositing pheromones. The specific pheromone left depends on the task being performed by the individual. Pheromones degrade and diffuse over time. They also aggregate as shown in figure 1. In this way, multiple interactions can be combined automatically to provide a single information source describing the aggregate state of the environment. Stigmergy expresses the synergy that occurs when multiple agents form a feedback loop with their environment.

The presence of pheromones in the environment provides dynamic information that regulates individual behaviors. Individual actions aggregate into macro-behaviors and pheromone signals aggregate into macro-information. In this way an agent modifies its environment, and the environment adapts to the needs of the agent.

The best-known example of this is foraging for food by ant colonies. Dorigo has expanded the basic concept into a general optimization methodology [12]. Each ant in a colony

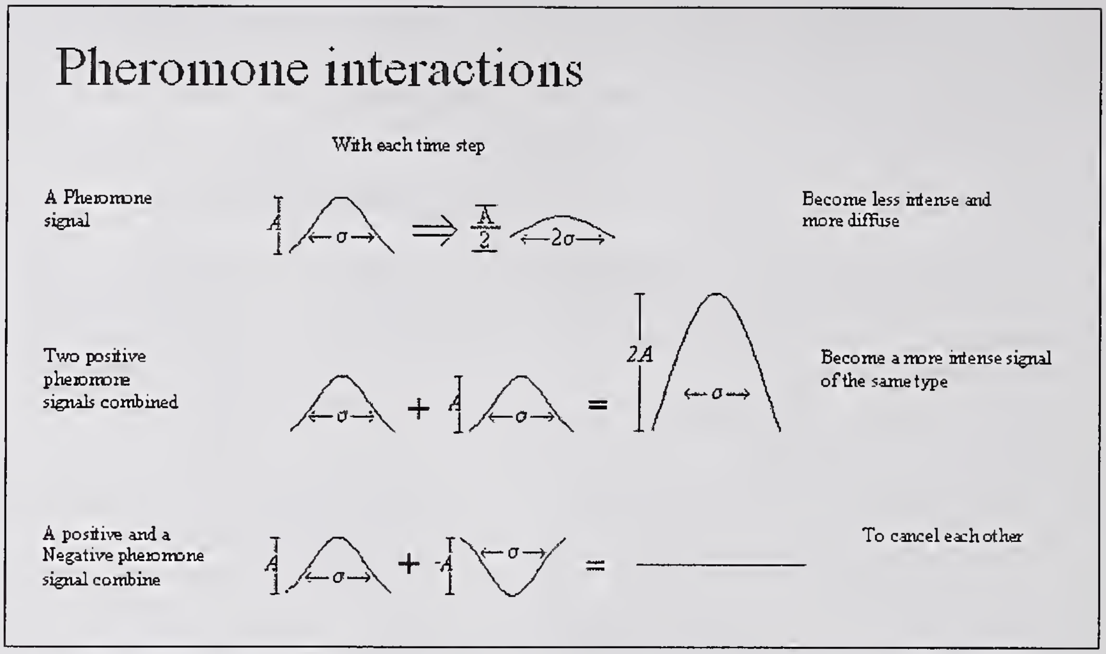

Pheromone interactions
With each time step
A Pheromone signal
A
σ
A/2
2σ
Become less intense and more diffuse
Two positive pheromone signals combined
A
σ
2A
Become a more intense signal of the same type
A positive and a Negative pheromone signal combine
A
-A
σ
To cancel each other

Figure 1 – Pheromone primitive characteristics for information diffusion and reinforcement

performs two basic behaviors, regulated by two basic pheromones:

- *Look for food* – wander in a stochastic manner depositing a "searching for food" pheromone. If the "found food" pheromone is detected, the stochastic movement is weighted to favor movement towards the "found food" pheromone.
- *Bring food to the nest* – when food is located, the ant picks it up. As long as the ant carries food it deposits the "found food" pheromone. It moves in a stochastic manner weighted towards the direction with the strongest "searching for food" pheromone signal.

In [12] and [5] this behavior is analyzed in detail. By heading towards the strongest concentrations of the pheromones, ants tend to follow the direct path. By allowing stochastic deviations, premature convergence to sub-optimal solutions is averted. By aggregating behaviors of many individuals, the system achieves a large degree of robustness.

Note that random decisions play a large role in this behavior. This resembles the use of random initial conditions in neural networks. Many other meta-heuristic approaches, such as genetic algorithms and simulated annealing [7], rely on stochastic, non-deterministic choices to find good quality results.

4. Synthetic Ecosystems

Self-organizing systems are characterized in Bonabeau [3] by:

- Creation of spatio-temporal structures in initially homogeneous media.
- Co-existence of many possible and reasonable solutions.
- The existence of bifurcations; common in non-linear systems [2].

Self-organizing systems of this type have several appealing aspects, such as robustness and conservation of resources. The existence of multiple possible solutions means that if one solution becomes untenable another can be found. Basing behavior on local decisions using purely local information reduces latency and bandwidth consumption. For these reasons a number of artificial systems have been designed using these principles.

Synthetic stigmergy has been applied to distributed route planning [5], military command and control [23], factory workflow design [9], and telecommunications networks [4]. It provides a convenient formalism for expressing dynamic interactions of multiple agents.

All of these approaches construct a synthetic environment for cooperating agents. Agents change the environment, adding information to it in the form of pheromones. Specific attributes of the pheromone such as speed of dissipation, diffusion rate, and meaning are specific to the individual application. Multiple simple agents then use the information aggregated by the environment to steer their partially stochastic behaviors. Note the similarity between this approach and the CA formalism. Macroscopic interactions between simple individual components provide complex adaptive behaviors.

5. Autonomous Sensor Networks

One application of this approach is in sensor networks. Distributed sensor networks use multiple autonomous sensor nodes to provide a sensing system with greater precision and dependability than any component sensor nodes [7]. When multiple sensor nodes survey the same region, redundancy reduces system sensitivity to single points of failure. At any point in time, a single sensor provides a single data point.

Collaborative signal processing aggregates data points into a more reliable global estimate with dependability estimates. This is similar to using multiple experiments to statistically determine a parameter value and its variance [20]. A number of military and commercial applications exist for this technology [6, 13].

Sensor nodes do local processing and relay information among themselves. They self-organize into a coherent whole, forming an ad hoc multi-hop network. Data is relayed from one node to another. Routing choices can be made dynamically using self-organization primitives such as pheromones. Master nodes determine a frequency-hopping schedule that slave nodes follow. Data can be forwarded from one cluster of nodes to the next, until a gateway to the Internet is reached, at which point, a number of user workstations can access the information simultaneously.

Sensor networks have a number of unique aspects. Manual deployment and placement of a large network of sensors would be time consuming and expensive. Ideally the nodes could be deployed automatically. When the number of nodes increases beyond a trivial number, manual network organization becomes problematic as well. Figure 2 illustrates many of the factors that influence network organization and deployment.

When there are a large number of nodes, manual task distribution becomes onerous and time consuming. If conflicts exist in the needs of different user communities the process becomes even more challenging. All of these reasons point to the fact that the networks must be capable of self-organization and autonomous tasking.

Nodes have a finite lifetime, which is shortened by computation, sensing, data transmission, and data reception because they are battery-powered. Most distributed dependability theories are irrelevant to these networks [15]. Distributed dependability verifies the properties of*safety* (lack of undesirable events in the network) and *liveness* (a networks eventual return to a long-term steady state). Since batteries will eventually be exhausted, the network will eventually fail. The property of liveness is impossible to attain. Instead, the system must strive for adaptability. It should reconfigure and tolerate multiple faults. Routing algorithms should avoid creating "hot-spots" that frequently relay data through the multi-hop network, since they will fail much more quickly than the rest of the network. Traffic to relay system housekeeping information should be kept to an absolute minimum.

Traffic patterns for sensor networks differ from those in more traditional *ad hoc* mobile communications networks, such as cell-phones. In traditional *ad hoc* networks, communications are desired between two specific nodes (customers). The network routing protocol needs to find the node no matter where it is located in the network. Sensor networks have the opposite task. Information is required about a specific location. The node identity is irrelevant. For this reason, routing is data-centric.

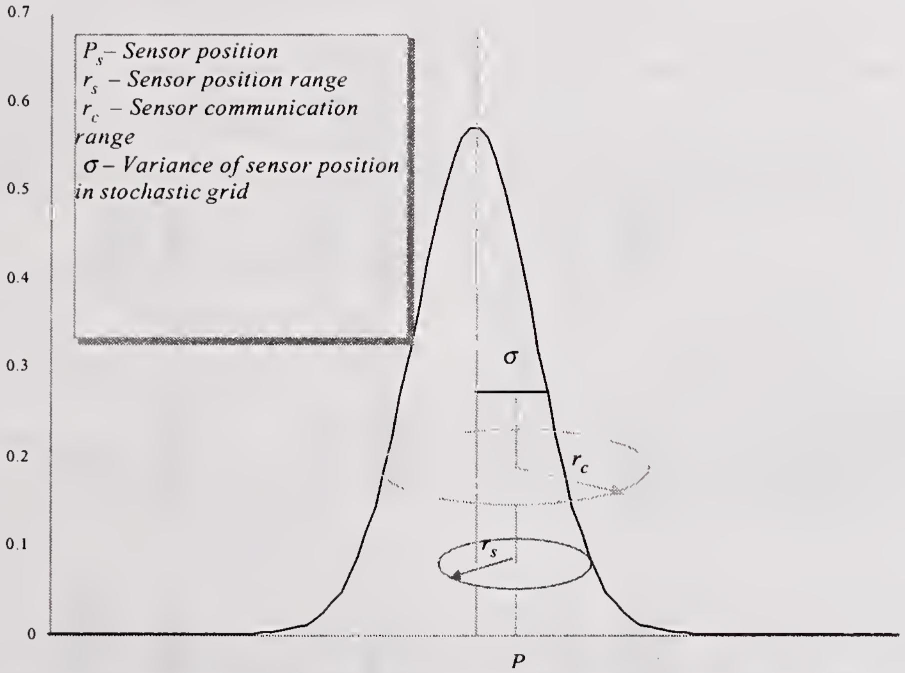

0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
P_s– Sensor position
r_s – Sensor position range
r_c – Sensor communication range
σ– Variance of sensor position in stochastic grid
σ
r_c
r_s
P

Figure 2. When networks are manually or autonomously deployed and configured, a number of factors need to be considered. These include sensor range r_S and communications range r_C. In the current implementation $r_s > r_C$. In addition to this the nodes position is generally known from GPS units and have an associated uncertainty

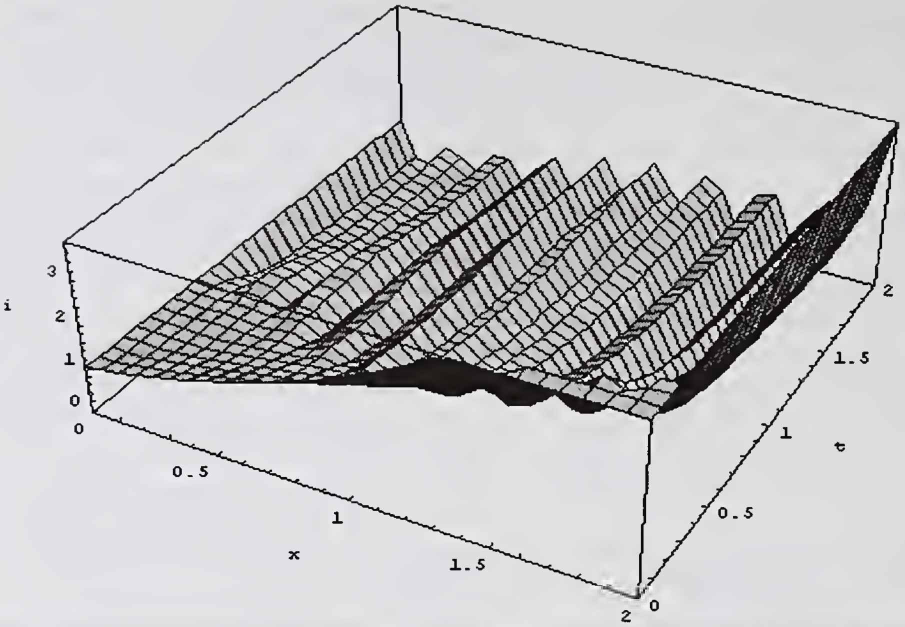


Figure 3 An example transient effect modeled for packet density in a simple sensor network.

In cell-phone networks, communications occur between nodes in the network for a conversation of unknown length. Conversation length is often modeled as an exponential distribution. A reasonable goal is to find a path through the network, which asymptotically approaches the least cost path over time. In sensor networks, queries tend to be punctual. They are either to inform the user of the current state, or inform the user quickly when a given event occurs. It will not be unusual for a single packet to be sufficient. Asymptotic optimality is irrelevant. Transient effects that can be ignored in other systems become much more important. Figure 3 illustrates an example of the transient effects modeled for a simple network.

The Reactive Sensor Network project at the Pennsylvania State University Applied Research Laboratory implements a mobile code infrastructure that augments sensor network adaptability [8]. This approach is inspired by the *active network* paradigm [26].

This approach helps in implementing a self-tasking network. Specific node work assignments need not be known in advance. The software can be reconfigured and modified as needs arise. Similarly if the battery fails on a node performing an essential task, another node can download the software needed to replace it. Figure 4 provides a view of how the individual nodes interact to form a single multicomputer. Notice that it is a macroscopic multicomputer aggregating the behaviors of its autonomous components.

Pheromone based control is also possible. One candidate pheromone is remaining battery power. Another candidate pheromone is distance to an Internet gateway. Combining the

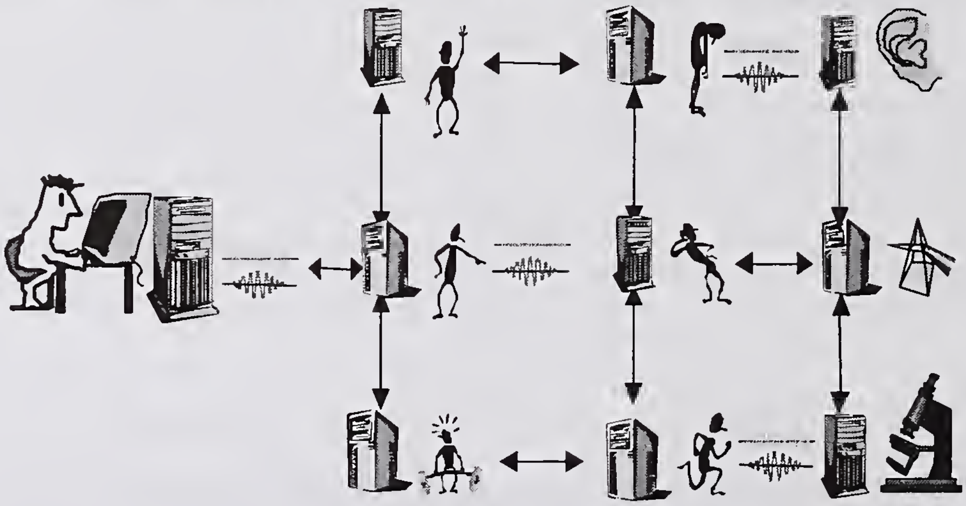


Figure 4. The network is a large computing system formed of individual nodes and sensing devices. Task distribution is determined based on current workloads.

two pheromones provides a self-organizing sensor data network synthetic eco-system that avoids the creation of "hot-spots." This extends the useful lifetime of the network. The CA formalism is useful in exploring a distributed system like autonomous sensor networks (figure 4). Behaviors of individual nodes can be simple and guided by local information. What is important is that the entire system develops complex adaptive behavior from interactions among the nodes.

6. Conclusion

Unfortunately, most intelligence metrics are inherently subjective. They often translate into the Supreme Court's metric for pornography: "I know it when I see it." Equally unfortunately, most of us recognize intelligence mainly when looking in the mirror. An example of this type of subjective and narcissistic metric is the Turing test. For the concept of intelligence to be useful, it needs an objective metric.

Can a purely deterministic system be considered intelligent? If this is the case, arithmetic equations and statements of fact are legitimate candidates for intelligence. To the contrary, intelligence is beyond rote memorization and execution of explicit recipes. Intelligence has a creative aspect. An intelligent entity must provide unexpected, creative, appropriate, results. This implies a nondeterministic, random, or stochastic component. Distributed networks of simple interacting automata are robust examples and are capable of performing general computations [27].

Two existing qualitative hierarchies provide objective metrics of intelligence:

- **Chomsky's language hierarchy:** (1) regular grammars recognized by finite state automata, (2) context free grammars recognized by push-down automata, (3) context sensitive languages recognized by linear bounded automata, and (4) recursively enumerable sets recognized by Turing machines [18].
- **Wolfram's complexity classes of CA's:** (1) stable, (2) repetitive, (3) chaotic, and (4) interesting.

To measure the IQ of intelligent systems another qualitative scale is needed that measures systems interactions with their environment:

- **Nonadaptive** – most systems
- **Adaptive** – can regulate parameters to fit environmental conditions. Most controllers would be in this class.
- **Self-Organizing** – adapt to their environment and autonomously reorganize as required. [12] and [8] are examples of this class.
- **Full stigmergy** – modify the environment to suit their goals. Nest building termites, wasps, and humans are in this category.

Discussing intelligent systems presupposes that intelligence is not a purely human attribute. It is an attribute in both living and artificial systems. For that reason, it is appropriate to use concepts from biological studies of non-human intelligence. In simulations of insect societies and construction of artificial systems, stigmergy has been the key to designing robust, creative, emergent behaviors. An appropriate metric for comparing intelligence should be based on the system's stigmergy, stigmergy being the system's ability to interact with and modify its environment to advance the system's goals.

7. Acknowledgments

This effort is sponsored by the Defense Advance Research Projects Agency (DARPA) and the Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-99-2-0520 (Reactive Sensor Network). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the author's and should not be interpreted as necessarily represent the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency (DARPA), the Air Force Research Laboratory, or the U.S. Government.

8. References

[1] Adami, C., *Introduction to Artificial Life*, Springer Verlag, New York, 1998.

[2] Alligood, K.T., Sauer, T.D. and Yorke. J.A., *Chaos: an introduction to dynamical systems.* Springer Verlag, New York. 1997.

[3] Bonabeau, E., Theraulaz, G., Deneubourg, J-L, Aron, S. and Camazine, S. "Self-organization in social insects," *Working Papers of the Satna Fe Institute 1997.* http://www.santafe.edu/sfi/publications/Working-Papers/97-04-032.txt, 1997.

[4] Bonabeau, E., Henaux, F., Guerin, S., Snyers, D., Kuntz, P., and Theraulaz, G., "Routing in Telecommunications Networks with 'Smart' Ant-Like Agents," *Working Papers of the Satna Fe Institute 1998.* http://www.santafe.edu/sfi/publications/Working-Papers/98-01-003.ps. 1998.

[5] Bonabeau, E., Dorigo, M., and Theraulaz, G., *Swarm Intelligence: From Natural to Artificial Systems.* Oxford University Press, New York 1999.

[6] Bonnet, P., Gehrke, J., Mayr, T., and Seshadri. P, Query Processing in a Device Database System. Ncstrl.cornell/TR99-1775. http://www. ncstrl.org, 1999.

[7] Brooks, R. R. and Iyengar, S., *Multi-Sensor Fusion: Fundamentals and Applications with Software.* Prentice Hall PTR, Upper Saddle River, NJ. 1998.

[8] Brooks, R.R., et al. "Reactive Sensor Networks: Mobile Code Support for Autonomous Sensor Networks," *Distributed Autonomous Robotic Systems DARS 2000.* Accepted for Publication. Springer Verlag. October 2000.

[9] Brueckner, S.A., *Return from the Ant.* Ph. D. dissertation. Humboldt University, Berlin. 2000.

[10] Davalo, E. and Naim, P., *Des Reseaux de Neurones.* Editions Eyrolles, Paris, 1989.

[11] Delorme, M., "An introduction to cellular automata," *Cellular Automata: a Parallel Model.* M. Delorme and J. Mazoyer (eds). pp. 5-50. Kluwer Academic Publishers, Dordrecht. 1999.

[12] Dorigo, M., Manniezzo, V. and Colorni, A., "The Ant System: Optimization by a colony of cooperating agents," *IEEE Transactions on SMC-Part B,* vol. 26, no. 1, pp. 1-13., 1996.

[13] Estrin, D., et al. Next Century Challenges: Scalable Coordination in Sensor Networks. ACM MobiCom 99, Seattle, WA, 1999.

[14] Free, J., *Pheromones of Social Bees*, Chapman and Hall, London, 1987.

[15] Gaertner, J., Fundamentals of Fault-Tolerant Distributed Computing in Asynchronous Environments. ACM Computing Surveys, Vol. 31, no. 1, 1-26, 1999.

[16] Grassé, P.P., "La reconstuction du nid et les coordinations inter-individuelles chez Bellicositermes Natalensis et Cubitermes sp. La theorie de la stigmergie: essai d'interpretation du comportement des termites constructeurs," *Insect Sociology*, vol. 6, pp. 41-84, 1959.

[17] H. Gutowitz, *Cellular Automata: Theory and Experiment*, MIT Press, Cambridge, MA, 1991.

[18] Hopcroft, J.E., and Ullman, J.D., *Introduction to Automata Theory, Languages and Computation,* Addison-Wesley, Reading, MA. 1979.

[19] Minsky, M., *The Society of Mind.* Simon and Schuster, New York. 1986.

[20] Neter, J.,Wasserman, W. and Kutner, M.H., *Applied Linear Regression Models*, Irwin, Burr Ridge, IL, 1989.

[21] Langton, C.G., ed. *Artificial Life: An Overview.* MIT Press. Cambridge, MA. 1996.

[22] Pandya, A.S. and Macy, R.B., *Pattern recognition with Neural Networks in C++.* CRC Press, Boca Raton, FL. 1996.

[23] Parunak, H. and Brueckner, S.,"Synthetic Pheromones for Distributed Motion Control," *JFACC Symposium in Advances Enterprise Control.* November 1999.

[24] Resnick, M., "Learning about Life," *Artificial Life.* Vol. 1, no 1-2, Spring 1994.

[25] Skirkevicius, A., "Some Characteristics of Insect Pheromone Communication," *Sensory Systems and Communications in Arthropods.* Pp. 55-61. Birkhaeuser Verlag, Basel. 1990.

[26] Tennenhouse, D. L,. J. M. Smith, W. D. Sincoskie, D. J. Wetherall, and Minden, G.J., A Survey of Active Network Research. *IEEE Communications Magazine,* vol. 35, no. 1, 80-86, 1997.

[27] Wolfram, S., *Cellular Automata and Complexity.* Addison-Wesley. Reading, MA. 1994.

Performance Self-Assessment by and for Regulation in Autonomous Agents

Elpida S. Tzafestas
Institute of Communication and Computer Systems
Electrical and Computer Engineering Department
National Technical University of Athens
Zographou Campus, Athens 15773, GREECE.
brensham@softlab.ece.ntua.gr

ABSTRACT

We are studying the problem of connecting intelligence to performance in the context of autonomous agents with very limited capabilities, where performance is suspected to include very few parameters and be easier to quantify than in more complex cases. We are reviewing and comparing three behavioral models that solve three typical autonomous agent problems, the explorer robot [1][2], the food-collector ant [3] and the cooperative tit-for-tat agent [4]. In all three cases and despite the apparent differences between them, we have defined a single problem-dependent performance measure and, on that basis, we have found that the most intelligent among several alternative models, i.e., the one that to the eyes of an observer achieves better performance, is a self-regulatory model involving a two-level regulatory process and an internal variable representing the state of the problem-solving process, thus self-assessing recent performance. The power of the agent lies in the second level and regulation mechanism, that is problem-dependent, and that has been shown to achieve the highest performance among many alternative models in all three problems. The whole design thus allows the agent to assess its actual performance and correct its behavior by modifying accordingly the first-level regulation rates, or equivalently by adapting the first level regulation law. From a symmetrical point of view, the agent may also be thought of as predicting the future state of the environment and adapting accordingly. The self-regulatory process appears therefore as both the means to effective performance assessment and the low-level prerequisite to enhanced intelligence.

1. INTRODUCTION : FUNDAMENTAL CONCEPTS

We are studying the problem of connecting intelligence to performance in the context of autonomous agents with very limited capabilities, where performance is suspected to include very few parameters and be easier to quantify than in more complex cases. By definition, the bottom-up study of intelligence relies on two axioms, evolution and interaction. The axiom of evolution states that higher forms of intelligence appear as a result of an evolutionary process that proceeds from simpler to more complex forms. Complex intelligence thus containts and requires antecedent simpler intelligence. On the other hand, intelligence has no absolute value, but depends on and is the result of dynamic interaction with a changing world. From an evaluation point of view, intelligence is not a well-defined nor a well-specified property, but it depends on an observer's point of view, or as Brooks [5] says "intelligence is in the eye of the observer". An agent demonstrating intelligence through dynamic interaction with a changing world has to be responsive to its environment and adaptive to a range of unpredicted events and situations. For the sake of enhanced stability, adaptivity methods should better be constructed or "controlled" by the agent itself. On the other hand, we, as designers of autonomous agents, are seeking universal design laws that will make our job easier in the long term. To this end, we are investigating a number of classical autonomous agents problems in an attempt to identify common design solutions, that is design solutions that share design principles.

In what follows, we are reviewing and comparing three behavioral models that solve three typical autonomous agent problems, the explorer robot [1][2], the food-collector ant [3] and the cooperative tit-for-tat agent [4]. In all three cases and despite the apparent differences between them, we have defined a single problem-dependent performance measure and, on that basis, we have found that the most intelligent among several alternative models, i.e., the one that to the eyes of an observer achieves better performance, is a self-regulatory model involving two regulatory processes and an internal variable representing the state of the problem-solving process, thus self-assessing recent performance.

The crucial internal agent variable has to be regulated within bounds. The goal of the agent is to either bring it to a limit (say 0) or prevent it from reaching the extremes. This design step depends on the definition of a quantitative environment or problem state, that will be next used as a metric to evaluate different design alternatives.

Regulation occurs using positive feedback, so that the agent's variable follows the tendency of the external world that it tries to represent, although the value of the variable almost never coincides with the truth, but rather it maintains a representational distance from it. At a second or meta level, another regulation process takes place that regulates the rates of the first level using negative feedback. The power of the agent lies precisely in the second level and regulation mechanism, that is problem-dependent, and that has been shown to achieve the highest performance among many alternative models (including the one without meta-regulation) in all three problems.

The whole design thus allows the agent to assess its actual performance and correct its behavior by modifying accordingly the first-level regulation rates, or equivalently by adapting the first level regulation law. From a symmetrical point of view, the agent may also be thought of as predicting the future state of the environment and adapting accordingly, because conceptually the negative regulation law is the following:

```
If (the world diverges from the agent's
   representation of it)
then (in the future) adapt so as to get
   closer to the world,
else (in the future) adapt so as to amplify
   differences from the world.
```

As a conclusion, the three case studies show that when an autonomous agent problem may be formulated as a regulation problem, the most intelligent alternative model, i.e. the one achieving highest performance, is one that continuously assesses its own performance and regulates its internal parameters accordingly. Therefore, in these cases intelligence appears as the result of low level self-evaluation and regulation.

2. CASE STUDY I : EXPLORER AGENTS

2.1. *The Problem*

A typical problem encountered in the behavior-based robotics literature is that of ***exploration*** : a set of agents (robots) lands on a planet with the mission to explore its surface for samples of minerals having certain properties. The robots arrive in a spaceship that serves as the planetary base in the course of the mission. The mission is accomplished when the whole surface contained within a certain distance from the base is explored, i.e., when the agents have "swept" the whole area and exhausted the sources of interest (cf. for instance [6][7]). The agents are supposed to return to their base once their mission is accomplished.

The exploration problem has been traditionally tackled from a "functional" point of view : "How does one or more agents sweep a delimited area to exhaust the sources of interest ?". The answer to this question is a control system, an architecture, that allows an agent to navigate, perceive, detect minerals etc., in order to sweep the area in question. A solution such as those encountered in the literature (for instance [8]) that comprises a random component and even without spatial reasoning or learning, statistically ensures the coverage of the interest field and the exhaustion of the mineral sources.

However, from a more "cognitive" point of view, this functionality alone does not respond to the crucial question : "***How do the agents know that they have swept the whole area, or that they have accomplished their mission ?***". In order to answer to that question, we have to reformulate the description of the sweeping task, in a way so as to include an expression, analytical or other, that represents the termination criterion, that is the exhaustion of the mineral sources. To this end, it is sufficient to define an environmental variable, the density of mineral sources, which characterizes the state of the explored area at any moment. In what follows, this density will be denoted as p_w. The explorer-sweeper agent's goal becomes therefore to bring the value of that variable to 0. We will see that an agent having a representation of that variable constitutes a simple solution to this description problem.

Lastly, we seek an agent model that would "optimize" performance, i.e. that would allow an agent to accomplish its mission as fast as possible.

In our simulations, the world under exploration is defined as a square around the central base : the size of the world is therefore the length of the square's edge (the results reported have been obtained in a 25x25 world). The agent's basic control system, as well as the simulation details, is given in [1][2]. We are analyzing next the single agent case, whereas the multiple agents case is studied in [1][2].

2.2. *The Solution : Reformulation of the Problem*

We come now to the second question : "How does the agent know it has swept the whole area in order to return to the base ?". It needs a way to detect the degree of task completion or else a termination criterion (sweeping completed). The only parameter of the task that can be useful to the development of a termination criterion is the source density in the world $p_w(t)$. If the agent knew in advance its initial value $p_w(0)$, we could define as termination criterion a formula such as *{$p_w(0)$ * sqr(r) samples have been collected}* (where *r* is the size of the square's edge, here 25). However, this criterion is not robust because if a sample is not detected, the agent will never terminate (on the other hand, we could certainly allow ourselves to miss a couple of samples).

A simple solution to this problem is to estimate continuously the value of $p_w(t)$ and, given that it falls to 0 as a side effect of the agent's activity, take as a termination criterion $p_w(0)=0$. Estimation of the value of $p_w(t)$ involves then a representational variable which is local to the agent ($p_a(t)$) and may be done through a simple formula of proportional adaptation :

```
Representational variable : $p_a(t)$
Proportional adaptation (window w, rate r) :
$p_a(t) = p_a(t-w) + diff * r$
$diff = p_{comp} - p_a(t-w)$
$p_{comp}$ = number of picked samples / number of
    moves (during the adaptation window)
Termination criterion :
    $p_a(t) < e_p$
    where $e_p$ is a small threshold (here,
    $e_p=0.001$)
```

The p_{comp} is the agent's estimate of p_w as computed during the adaptation window and the proportional law ensures that the estimate's update does not take place too quickly. This representation/adaptation system shows the advantage of robustness in front of perturbations/manipulations such as reinitialization of $p_w(t)$ during sweeping. Figure 1 illustrates the coevolution of the two variables $p_w(t)$ and $p_a(t)$. As is shown in the figure, ***the representational variable allows the agent to always solve its termination problem without ever taking the real value of the variable it represents*** (except a crossing point). Both variables fall progressively to 0 without ever taking the same value — we could say that $p_a(t)$ "follows" $p_w(t)$. Actually, the rapid rise of $p_a(t)$ in the beginning of the sweeping phase is due to the presence of a sensor of distant samples that makes the agent head toward the mineral sources minimizing its erratic behavior in a way that most of the visited places contain samples. The value of $p_a(t)$ falls then because the value of $p_w(t)$ decreases as a side-effect of the agent's activity who finds less and less samples.

2.3. *On Efficiency : Meta-Regulation*

Next, we proceeded to study the relation between the adaptation system's w and r parameters and the initial world value $p_w(0)$. The system has been simulated for several values of w and r in several initial world densities. The simulation results for three sets of adaptation parameters (quick, medium or slow adaptation) in a medium initial world density are given in fig. 1.

The quick adaptation is more operational than the medium one, which is in turn more operational than the slow one (always according to the task duration criterion). However, the quicker the adaptation, more fluctuations it shows, and the slower the adaptation, more delays it shows. Furthermore, the same parameter setting gives different results in different world densities : the difference in the results is reflected on the shape of the curves (for more curves, refer to [1][2]). More particularly, the agent's response to different perturbations (the shape of the curve of $p_a(t)$) differs according to the boundary condition ($p_w(0)$) : for the same parameter setting, the agent finishes its task more or less quickly according to the value of $p_w(0)$, that is the duration of the interval between the moment of picking of the last sample and the definitive return of the agent to the base is very variable. It seems therefore that to ensure the agent's operationality in different worlds, we need to find a means to combine the operational advantages of quick adaptation with the advantages of slow adaptation as far as curve regularity is concerned.

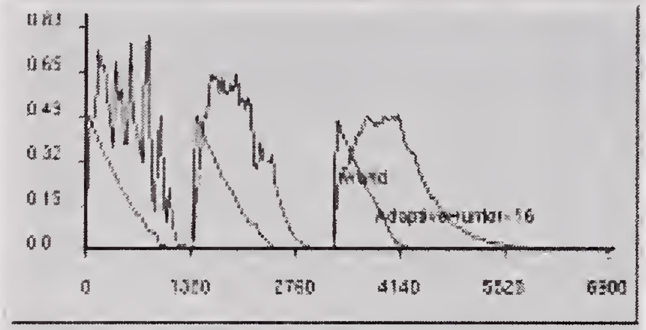


Figure 1. Performance of the agent for different parameter settings in a medium initial world density, $p_w(0)$=0.5 ($p_a(0)$=0.15), t_1=1437, t_2=3278, t_3=6821.
First part (quick adaptation) : w=15, r=0.3, dt_1=1437.
Second part (medium adaptation) : w=30, r=0.2, dt_2=1841.
Third part (slow adaptation) : w=45, r=0.1, dt_3=3543.

More precisely, we need a quick adaptation near the end (to terminate quickly), but a slow adaptation during picking (to avoid fluctuations). We have then to find a way to stabilize to the right parameter setting *on-line*. Otherwise stated, ***we need a meta-adaptation system***.

Meta-adaptation has to affect the w and r parameters in a way that adaptation becomes quicker when p_{est} is sufficiently close to $p_a(t)$ and slower when it is far from it. This meta-adaptation law translates the fact that the world is more reliable when it is not much different from the agent's idea about it, otherwise it should not be taken too seriously.

```
Meta-adaptation :
If $|diff|$ (= $|p_{comp}-p_a(t-w)|$) $\leq f_p$,
then quicker adaptation
  $r \rightarrow r_{max}$, $w \rightarrow w_{min}$
  ($r = r + r_r * (r_{max}-r)$, $w = w + r_w * (w_{min}-w)$)
otherwise slower adaptation
  $r \rightarrow r_{min}$, $w \rightarrow w_{max}$
  ($r = r + r_r * (r_{min}-r)$, $w = w + r_w * (w_{max}-w)$)
```

Figure 2 gives the results of applying the meta-adaptation system in three initial world densities ; as is shown in the figure, the agent's response (the shape of the curve) is the same for all three exemplary densities, or else the residue of mission duration after picking the last sample is approximately the same in all three cases.

We have shown in [1][2] that the operationality of the agent with the meta-regulation law does not depend qualitatively on the values of w_{min}, w_{max}, r_{min}, r_{max}, r_w and r_r. Furthermore, the initial condition ($p_a(0)$) plays no role either.

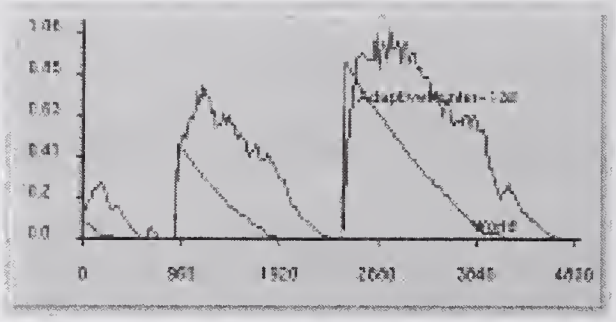


Figure 2. Performance of the agent with a meta-adaptation system for three initial world densities, low ($p_w(0)$=0.1), medium ($p_w(0)$=0.1) and high ($p_w(0)$=0.9) ($p_a(0)$=0.15). dt_1=897, dt_2=1663, dt_3=2211 (t_1=897, t_2=2560, t_3=4771). (f_p=0.1, w_{min}=15, w_{max}=40, r_{min}=0.15, r_{max}=0.3, r_r=r_w=0.2)

3. CASE STUDY II : TRAIL-MAKING ANTS

3.1. *The Problem*

In another variant of the previous problem ([8][9][10]) there are a few large sources distributed in the world. The solution in this case consists in allowing a robot to lay down trails or "crumbs" while carrying a source sample to the home base, that another robot or itself may follow to arrive to the source quickly. A different version of the problem considers that trails laid down by the robots evaporate slowly, in the same way as pheromone quantities laid down by real ants in the physical world ([11]).

The first complete solution has been given in [10], where a number of increasingly complex and increasingly satisfactory solutions have been analyzed. The Tom Thumb robot is able to successfully build, reinforce and correctly use trails from the home base to the source, while the Docker robot [10] uses an additional mechanism of sample "theft" from neighbors, which allows robots to build chains resembling harbor Dockers. The motivation for our work has been our feeling that the Tom Thumb robot as defined is not stable because it assumes unbounded numbers of "crumbs", which is not physically possible, and which would show in a real robotic implementation. A detailed presentation of what follows may be found in [3].

The Tom Thumb robot's behavioral diagram as described in [10] is as follows :

```
If (carrying samples)
     If (back home) lay down samples
     Else {go home, lay down 2 crumbs}
Else
  If (found samples) pick up samples
  Else
     If (crumb or stimulus sensed) (*)
        {follow stimulus, pick up 1 crumb}
     Else move randomly
```

```
(*) In the Docker robot, the condition
(crumb or stimulus sensed) is replaced
by (crumb or stimulus or loaded robot
sensed).
```

The Tom Thumb robot lays down two crumbs while homing, and picks up one crumb while following crumbs or stimuli. Unless otherwise stated, all simulations reported below use a 30x30 grid world with a large source at one of the corners and a population of 10 robots starting with 50 crumbs each. Robots may sense a sample or crumb from a distance of up to 3 grid cells.

We have simulated the behavior of the system as is, by measuring the quantities of crumbs deposited in the world or owned by individual agents. As was expected, the quantities of crumbs owned by robots generally fall below zero, while the quantity of crumbs deposited in the world may rise without limit. The exact values of these quantities depend on the problem parameters (distance from source to home base, number of robots and source size) that define the expected number of robot trips source-base necessary to complete the task.

3.2. *The Solution : Reformulation of the Problem*

An apparent question arising at this point is, "what if we just constrain robot behavior so as not to lay down crumbs when it does not have any ? aren't crumbs deposited so far enough ?" We have been able to see in several experiments that, first, depending on the problem parameters, the total quantity of crumbs might not be sufficient, in which case the path to the source will be disconnected, and, second, when it is sufficient — for instance if we start the above experiment with 1000 crumbs per agent — the total number of crumbs deposited in the world may rise tremendously. This last condition generates an important problem : the robots will continue being attracted for a long time to an empty source, that is, the surplus crumbs will be misleading. This observation brings us to the actual formulation of the above trailing problem :

We are seeking a laydown-pickup mechanism such that a trail to a source is built quickly and reinforced while the source exists and vanishes shortly after the source is exhausted.

The problem of agent crumb exhaustion lends itself to a simple solution. Every time a robot needs to lay down or pick up crumbs, it should do it in a way so as to preserve its own quantity of crumbs within some desired bounds $crumbs_{min}$ and $crumbs_{max}$, by using the following laws :

```
For laydown    crumbs(t+1) = crumbs(t) +
               $r_l$ * ($crumbs_{min}$ - crumbs(t))
For pickup     crumbs(t+1) = crumbs(t) +
               $r_p$ * ($crumbs_{max}$ - crumbs(t))
```

This simple regulation mechanism ensures that no agent will ever run out of crumbs completely. However, the absolute (real-valued) quantity of crumbs deposited or collected at each cycle will depend on the state of the agent : an agent with many crumbs will lay down more and pick up less than an agent with just a few crumbs remaining. This arrangement allows for trails to be built rapidly (because agents in the beginning tend to lay down large quantities of crumbs) and to vanish quickly (because agents toward the end of the task have statistically only a few crumbs, so they tend to pick up large quantities of crumbs). In what follows it will be assumed that $crumbs_{min}$=10 and $crumbs_{max}$ = 100, for all agents.

3.3. *On efficiency : Meta-Regulation*

A large laydown rate will be beneficial in the start and middle of the task, when the agents would like to build and reinforce a trail quickly, while a large pickup rate would be beneficial toward the end of the task, when the agents would like to destroy the trail to the exhausted source as quickly as possible. While a given parameter setting would be more desirable than another one in a particular context, our goal as designers should be to ensure the better behavior *globally*, i.e., to ensure that the system will "discover" or identify the proper parameter setting in each situation.

Consequently, what we really want is *not* a particular parameter setting, but a mechanism that will allow a robot to lay down more and pick up less crumbs at the beginning of the task (so as to build and reinforce the path) and vice versa toward the end (so as to destroy it quickly). To this end, a measure of the state of the task must be available. The only such measure that a robot may have is the number of the crumbs in the world. However, since this quantity cannot be directly perceivable, we have used an estimate of it, simply the number of crumbs at the current position of the robot. This estimate is used as follows :

```
For laydown
If crumbs(t) >= world_crumbs_estimate
    $r_l(t+1) = r_l(t) + r_{rl} * (r_{lmax} - r_l(t))$
else $r_l(t+1) = r_l(t) + r_{rl} * (r_{lmin} - r_l(t))$
For pickup
If crumbs(t) >= world_crumbs_estimate
    $r_p(t+1) = r_p(t) + r_{rp} * (r_{pmin} - r_p(t))$
else $r_p(t+1) = r_p(t) + r_{rp} * (r_{pmax} - r_p(t))$
```

As is obvious from the formulae, the rate of crumb laying increases when the robot owns more crumbs than may be found in its current position and decreases otherwise. Inversely, the rate of crumb picking increases when the robot owns less crumbs than may be found in its current position and decreases otherwise.

Figure 3 gives a typical result of the application of the above model. Surprisingly enough, the self-regulation of the laydown and pickup rates not just does change the qualitative behavior of the agents (the quantity of crumbs in the world rises quickly to a fairly high value, stays close to it during the task, and falls back quickly to zero when the source is exhausted, while showing far less fluctuations than in the previous case), but it improves results quantitatively as well : in all runs, including the one depicted, the duration of the task has been shorter than with the non-regulated model.

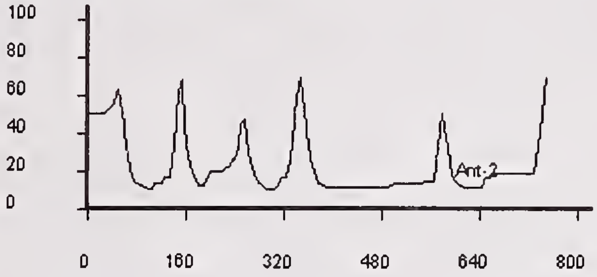


Figure 3. Quantity of crumbs owned by a meta-regulated agent in a typical run. It fluctuats between the upper and lower limits.

4. CASE STUDY III : ADAPTIVE TIT FOR TAT AGENTS

4.1. *The Problem*

A major issue on the intersection of artificial life and theoretical biology is cooperative behavior between selfish agents. The cooperation problem states that each agent has a strong personal incentive to defect, while the joint best behavior would be to cooperate. This problem is traditionally modeled as a special two-party game, the Iterated Prisoner's Dilemma (IPD).

At each cycle of a long interaction process, the agents play the Prisoner's Dilemma. Each of the two may either cooperate (C) or defect (D) and is assigned a payoff defined by the following table.

Agent	Opponent	Payoff
C	C	3 (= Reward)
C	D	0 (= Sucker)
D	C	5 (= Temptation)
D	D	1 (= Punishment)

Usual experiments with IPD strategies are either tournaments or ecological experiments. In tournaments, each strategy plays against all others and scores are summed in the end. In ecological experiments, populations of IPD strategies play in tournaments and successive generations retain the best strategies in proportions analogous to their score sums.

The first notable behavior for the IPD designed and studied by Axelrod [12] is the Tit For Tat behavior (TFT, in short) :

Start by cooperating,
From there on return the opponent's previous move.

This behavior has achieved the highest scores in early tournaments and has been found to be fairly stable in ecological settings.

The best designed behavior found so far in the literature is GRADUAL [13] which manages to achieve the highest scores against virtually all other designed behaviors. This behavior starts by cooperating and then plays Tit For Tat, except that it does not defect just once to an opponent's defection. Instead, it responds by playing blindly (nxD)CC, where n is the opponent's number of past defections. That is, GRADUAL responds with DCC to the first opponent's defection, DDCC to the second, etc. The justification given for the performance of this behavior is that it punishes the opponent more and more, as necessary, and then calms him down with two successive cooperations.

The motivation for our work has been our conviction that a behavior comparable to GRADUAL could be found, that has not permanent, irreversible memory. Instead, we are after a more adaptive tit-for-tat based model that would demonstrate behavioral gradualness and possess the potential for stability in front of changing worlds (opponent replacement etc.).

Before proceeding, let us examine the high scores that GRADUAL obtains against other behaviors. Designed behaviors found in the literature usually fall in one of three categories :

- Behaviors that use feedback from the game, usually cooperative behaviors unless the opponent defects, in which case they use a retaliating policy (tft, grim, gradual, etc.).
- Behaviors that are essentially cooperative and retaliating, but start suspiciously by playing a few times D in the beginning, so as to probe their opponent's behavior and decide on what they have to do next. For example, suspicious tft (STFT) and the "prober" behavior of [13].
- Behaviors that are clearly irrational, because they don't use any feedback from the game. For example, the random behavior and all blind periodic behaviors such as CCD, DDC etc.

A behavior will maximize its score, if it is able to converge to cooperation with all behaviors of the first two categories and converge to defection against behaviors of the third category. Steady defection against periodic behaviors is necessary in order to achieve the highest possible score (see [4], for details).

The GRADUAL behavior fulfills both of the above specifications, because it responds with two consecutive C's after a series of defections, giving the chance to STFT or prober behaviors to revert to cooperation, and converges to ALLD against irrational behaviors. A solution to the permanent memory problem has to demonstrate the same property.

4.2. *The solution : Reformulation of the Problem*

The adaptive behavior that we are seeking should be essentially tit-for-tat. Moreover, it should demonstrate fewer oscillations between C and D. To this end, it should have an estimate of the opponent's behavior, whether cooperative or defecting, and react to it in a tit-for-tat manner. The estimate will be continuously updated throughout the interaction with the opponent. The above may be modeled with the aid of a continuous variable, the world's image, ranging from 0 (total defection) to 1 (total cooperation). Intermediate values will represent degrees of cooperation and defection. The adaptive tit-for-tat model can then be formulated as a simple linear model :

```
Adaptive tit-for-tat
If (opponent played C in the last cycle)
    then
world = world + r*(1-world), r is the
    adaptation rate
else
     world = world + r*(0-world)
If (world >= 0.5) play C, else play D
```

The usual tit-for-tat model corresponds to the case of r=1 (immediate convergence to the opponent's current move). Clearly, the use of fairly small r's will allow more gradual behavior and will tend to be more robust to perturbations.

Now, let us simulate the behavior of the adaptive tit-for-tat agent against all three types of behaviors described earlier.

- For initially cooperative behaviors with feedback and a retaliation policy, the model cooperates steadily and converges quickly to total cooperation.
- For suspicious or prober behaviors, the model plays exactly like tit-for-tat, while the value of the world variable oscillates around the critical value of 0.5 (see figure 4 against suspicious tft).
- For periodic behaviors, the value of the world variable converges quickly to oscillations around the characteristic value of "number_of_C's/number_of_D's" in the opponent's period.

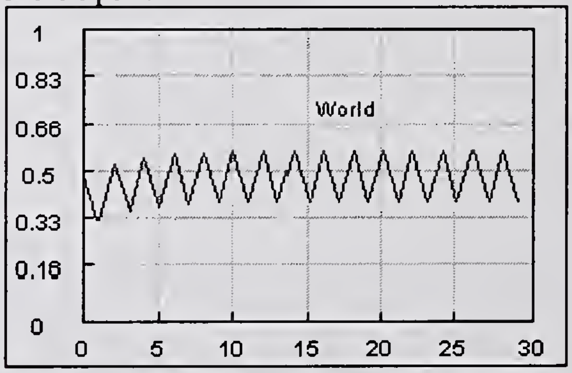


Figure 4. Interaction of adaptive tit-for-tat with suspicious tit-for-tat (r=0.2, world(0)=0.5).

4.3. *On Efficiency : Meta-Regulation*

It can be seen that the previous version of the model suffers from manipulation of the world variable by the opponent. This shows as stabilization of the agent to an oscillatory behavior (as is the case against stft) or a steady cooperative behavior against irrational agents (as is the case against CCD). To bypass this problem, we exploited our observation that different rates for cooperation and defection (r_c and r_d, respectively) yield different results. More specifically, we observed that the adaptive tit-for-tat agent manages to get opponents such as stft or the prober to cooperate if $r_c>r_d$, while it manages to fall to steady defection against periodic behaviors if $r_c<r_d$.

Thus, what we need at this point is a method for the adaptive tit-for-tat agent to discover whether the opponent uses a retaliating behavior or is just irrational and to adopt accordingly the proper rate setting. We have designed and examined several such variants for estimating the opponent's irrationality and we have finally found the following rule :

```
Throughout an observation window, record how
many times (n) the agent's move has
coincided with the opponent's move. At
regular intervals (every "window" steps)
adapt the rates as follows :
If (n>threshold) then
        $r_c = r_{min}$, $r_d = r_{max}$
else    $r_c = r_{max}$, $r_d = r_{min}$
The rule may be translated as :
If (the world is cooperative enough)* then
        $r_c = r_{min}$, $r_d = r_{max}$
else    $r_c = r_{max}$, $r_d = r_{min}$
(*) recall that "my move = opponent's move"
is the so-called pavlovian criterion of
cooperation ([14])
```

Note that the agent drops its cooperation rate when the world is assumed cooperative, and increases it otherwise, that is, it uses negative feedback at the rate regulation level.

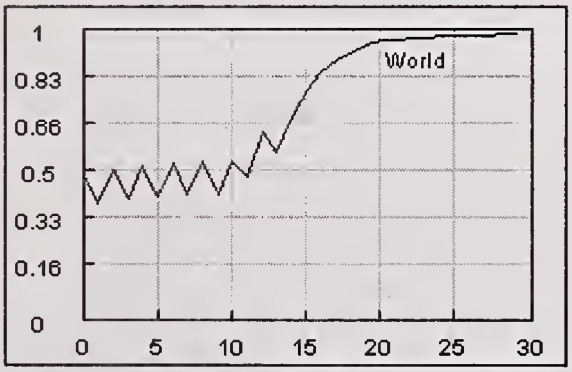


Figure 5. Interaction of the meta-regulated adaptive tit-for-tat agent with suspicious tit-for-tat ($r_c(0)=0.2$, $r_d(0)=0.2$, $r_{max}=0.3$, $r_{min}=0.1$, world(0)=0.5, window=10, threshold=2). Compare with figure 4.

We have shown in simulations that the adaptive tit-for-tat agent with the meta-regulation mechanism converges to the proper behavior against both retaliating and irrational agents. For example, figure 5 gives the behavior of the meta-regulated adaptive tit-for-tat agent against STFT.

Finally, the adaptive agent manages to differentiate between a retaliating agent and an irrational one that has initially the same behavior. The agent first assumes that the opponent is retaliating and becomes increasingly cooperative, but soon finds out that the opponent is actually irrational and reverts to defection.

5. DISCUSSION : ELABORATING THE CONCEPTS

In all three case studies, we have shown that the agent's behavior is based on a critical variable that drives its motivation to act. This variable is coupled with the environment through the agent's behavior. By regulating its own variable, an agent tries to regulate the corresponding world variable. Furthermore, this variable has *cognitive value*, since it represents the agent's idea about the state of the environment. Seen this way, the agent may be thought of as trying to approach or approximate the world variable, i.e., as trying to adapt to its environment. The regulated variables appear to be critical for an agent's survival or operationality, and correspond to what Ashby [15] called *essential variables*.

The operationality of the behavior is ensured through an additional self-regulation mechanism acting on the adaptation rates. This is an important observation, since it is compatible with the dynamical approach to cognition [16], stating that the most important factor in cognitive mechanisms is the nature of dynamics involved. Mechanisms like the ones developed here may be also regarded as a first step toward the realization of autopoietic systems :

> "… an autopoietic system is a homeostat … the critical variable is *the system's own organization …*" ([17], p. 66)

In sum, we have shown that self-assessment of performance by an agent is done with the aid of a double regulatory process and it allows it to become more operational in its work. This is in line with classical control theory, where regulatory mechanisms are used as the basis of behavior [18]. Inversely, similar regulatory processes may be designed for other problems, provided that the appropriate performance measure (or cognitive variable) and its assessment model are given or may be identified. In this sense, the long term perspective of this work is to build a regulation theory for reactive autonomous agents. To this end, a number of issues have to be investigated :

- How do we identify the critical cognitive variable in each case ? Equivalently, how do we formulate regulation in each case ?
- How many first level rates are necessary ? Equivalently, how many independent regulation processes are there ?

Recall that the explorer agent has one such rate, whereas both the trail-making agent and the adaptive tit-for-tat agent have two of them.

- Which is the meta-regulation criterion ? Note that in all three cases studied this criterion is purely qualitative and problem-dependent. Equivalently, this issue translates to "How can we observe and qualify a regulatory process ?".
- What is the nature of the meta-regulation dynamics ? A few initial experiments show that most probably a "bang-bang" dynamics (high-low value) is enough, because what counts is the relation between two rates rather than their absolute values.
- Finally, what is the role and value of "behavior in the empty" (without perturbation) ? This behavior is purely agent-specific and may differ among different agents, due to different parameter settings, defining thus the individual "character" of an agent. Initial experiments have shown that the behavior in the empty allows some limited prediction to be made.

As a general conclusion, the answers to questions such as the above could teach us a lesson on the power and potential of regulation mechanisms for apparently qualitative problems. They could also deepen our understanding of the scope and limits of such mechanisms amd prompt us to problems of immediately higher complexity, where regulation would not be enough and why this is so.

6. REFERENCES

[1] Tzafestas, E., *Vers une systémique des agents autonomes : Des cellules, des motivations et des perturbations*, Ph.D. diss., Univ. Pierre et Marie Curie, Paris, 1995.

[2] Tzafestas, E., "Regulation Problems in Explorer Agents", submitted.

[3] Tzafestas, E., "Tom Thumb Robots Revisited : Self-Regulation as the Basis of Behavior", *Proceedings Artificial Life VI*, San Diego, CA, June 1998.

[4] Tzafestas, E., "Toward Adaptive Cooperative Behavior", *Proceedings Simulation of Adaptive Behavior 2000*, Paris, France.

[5] Brooks, R.A., "Elephants don't play chess", in P. Maes (ed.), *Designing Autonomous Agents*, MIT Bradford Press, 1991.

[6] Brooks, R.A. and Flynn, A.M., "A robot being", *Robots and Biological Systems : Towards a new Bionics ?* (P. Dario, G. Sandini and P. Aebischer), 1989, pp. 679-701.

[7] Beckers, R., Holland O.E. and Deneubourg, J.-L., "From local actions to global tasks: Stigmergy and collective robotics", *Proceedings Artificial Life IV* (R. Brooks and P. Maes, Eds.), MIT Press, Cambridge, MA, 1994, 181-189.

[8] Mataric, M., "Designing emergent behaviors : From local interactions to collective intelligence", *Proceedings Simulation of Adaptive Behavior 1992*, 432-441.

[9] Steels, L., "Towards a theory of emergent functionality", *Proceedings Simulation of Adaptive Behavior 1990*, 451-461.

[10] Drogoul, A. and Ferber, J., "From Tom Thumb to the Dockers : Some experiments with foraging robots", *Proceedings Simulation of Adaptive Behavior 1992*, 451-459.

[11] Deneubourg, J.-L., Aron, S., Goss, S. and Pasteels, J.M., "The self-organizing exploratory pattern of the Argentine Ant", *Journal of Insect Behavior* 3(1990):159-168.

[12] Axelrod, R., *The evolution of cooperation.* Basic Books, 1984.

[13] Beaufils, B., Delahaye, J.-P., and Mathieu, P., "Our meeting with gradual: A good strategy for the iterated prisoner's dilemma", *Proceedings Artificial Life V*, Nara, Japan, 1996.

[14] Nowak, M., and Sigmund, K., "A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game", *Nature* 364(1993):56-58.

[15] Ashby, W.R., *Design for a brain - The origin of adaptive behaviour.* 2nd revised edition, Chapman & Hall, 1960.

[16] van Gelder, T., and Port, R., "It's about time : An overview of the dynamical approach to cognition", in *Mind as Motion : Explorations in the dynamics of cognition*, by T. van Gelder and R. Port, Cambridge, Mass.: MIT Press, 1995.

[17] Maturana, H.R., and Varela, F. (1980). *Autopoiesis and cognition — The realization of the living*, Dordrecht/ Boston, D. Reidel Publishing.

[18] Slotine, J.-J.E. (1994). Stability in adaptation and learning, *From animals to animats 3, Proceedings of the 3rd International Conference on Simulation of Adaptive Behavior*, MIT Press, pp. 30-34.

On Measuring Intelligence in Multi Agent Systems

Siva Perraju Tolety
GTE Laboratories
40 Sylvan Road,
Waltham MA 02451 USA
toletys@acm.org

Garimella Uma

Abstract

Intelligent behavior means doing the right thing [1]. Due to the bounded rationality of agents it is not always possible to do the right thing. Hence intelligence implies doing the best possible given the resources an agent or a multi agent system has. So, a measure of intelligence should reflect an evaluation of the process by which the agent or the multi agent systems arrive at exhibiting intelligent behavior. In multi agent systems, intelligent behavior is emergent in nature rather than additive. So, measures of intelligence should attempt to estimate the net resultant behavior rather than the individual fine grained reasoning processes of individual agents. Intelligent quotient measures for the human mind attempt to arrive at a single number based on a battery of tests. This number is a reflection of the individual's standing in his or her group. In this paper we attempt to present our ideas about arriving at such measures for multi agent systems.

1. Introduction

Intelligence is expected to allow an agent to do the right thing. The level of intelligence is reflected in the appropriateness of the actions undertaken by an agent in the given circumstances. Measures of intelligence have been used in the human society for a variety of purposes. These uses range from efforts to identify deficiencies in individuals and help them improve in these areas to efforts to rank people according to their capabilities in a given area.

Research in agents and multi agent systems is maturing and systems are being deployed in real world settings. Consequently, users of these systems would like to evaluate the system, understand its advantages and deficiencies and improve upon the same. Also if multiple intelligent systems purport to accomplish identical or similar tasks, the users of these systems will have a natural interest in making a comparison of the different systems. Similar needs in the human society gave birth to different performance measures. The measures are usually referred using the generic term *Intelligence Quotient (IQ).* These varied measures are based on differing views of intelligence. In Section 2, we briefly outline the differing views of human intelligence, and how these views lead to different perspectives of IQ measurements. In Section 3, we outline different multi agent architectures and highlight various aspects of these architectures that can be measured. In Section 4, we outline a possible measure for multi agent architecture developed for problem solving activities in real time domains. Section 5 presents the conclusions.

2. Intelligence and its Measurements in the Human Society

Intelligence is an abstract concept and reflects in some way the competencies and skills an individual possess. Some of the questions about intelligence include [2]:

- Is mental competence a single ability applicable in a variety of settings? or

- Is competence produced by specialized abilities, which a person may or may not possess independently?

If we can resolve this question about intelligence, the next question what are the metrics by which you can measures these competencies. Do these measures reflect in the every day problem solving ability of the individual? The answers to these questions depend to a large extent on the perspective; one subscribes to, about intelligence. Two popular views about intelligence are

- psychometric views and
- Cognitive-psychology view.

The psychometric view of intelligence places emphasis on scores obtained in carefully designed tests to evaluate specific skills. This view gives rise to the popular notion of intelligence quotient. Several versions of intelligent quotients exist. Usually these numbers are arrived by performing factor analysis on scores achieved on tests about different skills. Thus IQ measures reflect the level of what psychologists call *crystallized intelligence (Gc)*. Crystallized intelligence is the ability to apply previously acquired problem solving methods to the current problem. Measures of crystallized intelligence correlate strongly with another aspect of intelligence viz. *fluid intelligence (Gf)*. Fluid intelligence is the ability to develop techniques for solving problems that are new and unusual from the problem solver's perspective.

The cognitive-psychology view is that thinking is a process of creating mental representation of the current problem, retrieving information that appear relevant and manipulating the representation in order to obtain an answer [2]. This definition encapsulates the concepts of Gc and Gf. Forming a mental representation of the problem is akin to fluid intelligence and extraction of relevant information is similar to crystallized intelligence. A variety of tests based on the cognitive-psychology view of intelligence are also available.

3. Intelligence in Agents and Multi Agent Systems

Agent architectures reflect the underlying problem solving processes. Agents can be broadly classified as either reactive or deliberative. Reactive agents are usually preprogrammed to respond in particular ways to various stimuli from the environment. Agents with deliberative architectures usually use some reasoning process to arrive at a solution. Reflecting upon the classification of intelligence discussed in the earlier section, we can assume that reactive agents depend on crystallized intelligence while deliberative agents depend on fluid intelligence. So, psychometric based measures are appropriate for reactive architectures, while tests that try and evaluate the reasoning processes are appropriate for deliberative agents. Just as IQ tests target different groups of the society, tests in the agent world should also target specific agent sets. For example we might design a test suite that tries to evaluate spidering skills of Internet spider agents.

The power of multi agent systems is in their property of emergent behavior. Apart from the domain knowledge, multi agent systems possess some special features, which make them very attractive. These are

- communication and
- agent interaction

Agent interaction is achieved in a variety of ways. Coordination is the generic agent interaction mechanism that helps agents in a multi agent setting achieve their individual and common goals. Coordination can be achieved either through cooperative or competitive mechanisms. Communication protocols based on theories like speech acts enable agents to exchange small by semantically rich messages in aid of their problem solving. In this perspective of multi agent systems, the communication and agent interaction aspects of the systems seem to determine the intelligence or performance[1] of the system. Hence we propose that any performance measure for multi agent systems should in some way be able to rank different systems along these two dimensions. So, an IQ measure for multi agent systems is a function of three separate factors. They are

- domain knowledge(DK)
- individual agent reasoning capabilities (ARC)
- communication (COMM) abilities and
- efficacy of agent interaction AI).

MIQ = *f(DK,IARC, COMM, AI)*

Since we are interested in evaluating multi agent settings, we can ignore the factors that can be attributed to individual agents viz. DK and ARC. This implies we are assuming that all agents are equally capable in a given domain. This assumption though not suitable for rigorous measurements, could however be a good starting point. Hence

MIQ = *f(COMM, AI)*

4. Measuring intelligence / performance in TRACE

TRACE (Task and Resource Allocation in a Computational Economy) is a system of multi agent systems designed to operate under time constraints and load variations [3,4]. TRACE approach to problem solving is based on an adaptive organizational policy. The TRACE system is market based multi agent system. Tasks and resources are allocated to different multi agent systems based on their problem solving load and the price they are willing to pay for the resources. We assume that knowledge can be transferred among agents and thus domain knowledge plays no particular role in the evaluation of the multi agent systems.

Intuitively, we know that the efficiency of a player in a market is determined by how efficiently the multi agent system trades its funds for resources to aid in problem solving activities. Different multi agents systems in TRACE can adopt different policies to decide on their problem solving activities.

Now if we attempt to measure the performance of multi agent systems with different policies in the TRACE setting, what are the attributes that can capture the essence of the equations in the previous section? Multi agent systems sign up or commit for tasks and attempt to complete them. In this process they undertake both communication and agent interaction tasks. These tasks are time bound and task completion beyond a deadline is a wasted problem solving activity. In order to achieve maximum returns for the problem solving activity, multi agent systems will drop some tasks (decommit). The lower the number of decommitments the better the performance. The number of decommitments reflects how much a given system has overreached. It in turn reflects on the shortcomings in its communication, negotiation and problem solving abilities. Thus in the case of the TRACE system we feel that a normalized number of decommitments is an accurate measure of the performance of the multi agent systems.

We have implemented a prototype of the TRACE system, in which it is possible to introduce multi agent systems with different problem solving abilities and processes. We intend to formulate a simple problem to be solved by the multi agent systems. We then intend to measure the number of decommitments made and determine if this measure is a reasonably accurate measure of the performance of the multi agent system.

[1] We assume that higher level of intelligence results in better performance. Consequently, a MAS which is better at a given set of tasks than another MAS can be considered to be more intelligent.

5. Conclusions

In this paper we made an attempt at trying to understand the basis of intelligent measures used in the human society. Research in agent systems and multi agent systems led to the development of architectures that in some way try to mimic the problem solving skills in human beings. Thus the science of intelligent quotient measurements can be applied to the domain of intelligent agents and multi agent systems. In market based agent systems, we propose that the number of decommitments made by an agent along with the resources consumed is a measure of its ability. In more cooperative settings different but appropriate measures need to be designed. We conclude that the measures need to be designed by considering a family of agent architectures. For example we feel that a measure designed for a market based agent system will be ill suited for a cooperative multi agent system. We are currently experimenting with a multi agent system TRACE to understand the criteria for measuring its performance.

References

1. S. Russel and E. Wefald, Do the Right Thing, MIT Press, 1991
2. E Hunt, The Role of Intelligence in Modern Society, American Scientist, July- August 1995.
3. S Fatima, G Uma, T S Perraju, An Adaptive Organizational Policy for Multi Agent Systems, ICMAS 2000, Boston USA. July 2000.
4. S Fatima, G Uma, “AASMAN” An Adaptive Organizational Policy for a society of Market based agents, Sadhana, Academy Proceedings of Engineering Sciences, Indian Academy of Sciences, Vol 23, No. 4, pages 377-392.

On the Development of Metrics for Multi-Robot Teams within the ALLIANCE Architecture

Lynne E. Parker
Center for Engineering Science Advanced Research
Computer Science and Mathematics Division
Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, TN 37831-6355

ABSTRACT

Quantitatively evaluating the effectiveness of software architectures for multi-robot control is a challenging task. Exacerbating the problem is the fact that architectures are typically constructed to address different design goals and application domains. In the absence of benchmarks that capture the variety of issues that arise in multi-robot coordination and cooperation, the system developer can only evaluate an architecture for its own qualities. In this article, we summarize the metrics of evaluation that we utilized in applying our ALLIANCE architecture [17] to eight different application domains for multi-robot team control. We explore the implications of the metrics we have chosen and offer suggestions on future productive lines of research into metrics for multi-robot control architectures.

Keywords: *Multi-robot cooperation, metrics, ALLIANCE*

1 Introduction

Research work in multi-robot systems has progressed significantly in recent years. Issues that have been studied are diverse, and include task planning and control [1, 17, 12]; biological inspirations [6, 7, 13]; motion coordination [27, 2, 4]; localization, mapping, and exploration [22, 21]; explicit and implicit communication [5, 9]; cooperative object transport and manipulation [23, 25]; reconfigurable robotics [28, 24, 26]; and multi-robot learning [11, 12, 10]. Demonstrations have been given of multi-robot teams performing a variety of tasks, such object pushing, foraging, cooperative tracking, traffic control, surveillance, formation-keeping, and so forth.

However, most of this research is very specific and illustrates only one or two basic concepts per project. Comparisons across different methodologies are difficult and quantitative evaluations of various multi-robot control algorithms are scarce. While this is not unexpected for a field as new as cooperative robotics, enough progress has been made that we believe it is time to begin determining how we identify and quantify the fundamental advantages and characteristics of multi-robot systems. The characteristics most often cited for motivating the use of multi-robot teams are as follows:

- increased robustness and fault tolerance through redundancy,
- a potential for decreased mission completion time through parallelism,
- a possibility for decreased individual robot complexity through heterogeneous robot teams, and
- an increased scope of application due to tasks that are inherently distributed.

Other than direct measures of time, these characteristics are hard to quantify, yet vital to enabling the field to make objective comparisons and evaluations of competing architectures. Thus, much research is needed in this area.

2 Background

Measuring the performance of intelligent systems in general, and multi-robot systems in particular, is a much-understudied topic. Some beginning work has been accomplished by Balch [3], who has developed metrics for measuring multi-robot team diversity. However, little research has addressed the general issues of cooperation that provide guidelines for the quantification and selection of the appropriate cooperative team for any given set of mission specifications. Such a characterization would be a significant step towards the commercialization of cooperative systems, as it would facilitate the design of the appropriate cooperative team for a given application. Issues of particular interest in such a characterization include the following:

- Quantifying the overall system capability versus the system complexity,
- Determining the appropriate distribution of capabilities across robot team members for a given application,
- Ascertaining the most appropriate control strategy for a given robot team applied to a given application so as to maximize efficiency, fault tolerance, reliability, and/or flexibility, and

- Determining tradeoffs in control strategies in terms of desirable traits, such as efficiency versus fault tolerance.

Examples of this type of research include [8], which develops measures of effectiveness and system design considerations for the generic area coverage application, and [14], which compares the power of local versus global control laws for a "Keeping Formation" case study. However, much more work remains to be accomplished towards the development of quantitative comparisons of alternative approaches to cooperative team design. An understanding of the factors that influence the relative performances of various approaches to cooperative control will enable not only an evaluation of existing methodologies, but will also aid in the design of new cooperative control approaches.

Since addressing the issue of quantitative measurement and system integration for the entire field of cooperative robotics is extremely challenging, we have begun work in this area by focusing on our experiences with the ALLIANCE architecture. We developed the ALLIANCE architecture [17] to enable fault tolerant action selection in multi-robot teams. The focus was on an approach that operated successfully amidst a variety of uncertainties, such as sensory and effector noise, robot failures, varying team composition, and a dynamic environment. We have implemented ALLIANCE in eight different application domains in the laboratory. This experience is the basis for our beginning work in the development of general metrics and system integration as it applies to the use of ALLIANCE.

3 Brief Overview of ALLIANCE

We developed the ALLIANCE architecture to enable fault tolerant action selection in multi-robot teams. The focus was on an approach that operated successfully amidst a variety of uncertainties, such as sensory and effector noise, robot failures, varying team composition, and a dynamic environment. The ALLIANCE architecture, shown in Figure 1, is a behavior-based, distributed control technique. Unlike typical behavior-based approaches, ALLIANCE delineates several behavior sets that are either active as a group or are hibernating. Each behavior set of a robot corresponds to those levels of competence required to perform some high-level task-achieving function. Because of the alternative goals that may be pursued by the robots, the robots must have some means of selecting the appropriate behavior set to activate. This action selection is controlled through the use of motivational behaviors, each of which controls the activation of one behavior set. Due to conflicting goals, only one behavior set is active at any point in time (implemented via cross-inhibition of behavior sets). However, other lower-level competencies such as collision avoidance may be continually active regardless of the high-level goal the robot is currently pursuing.

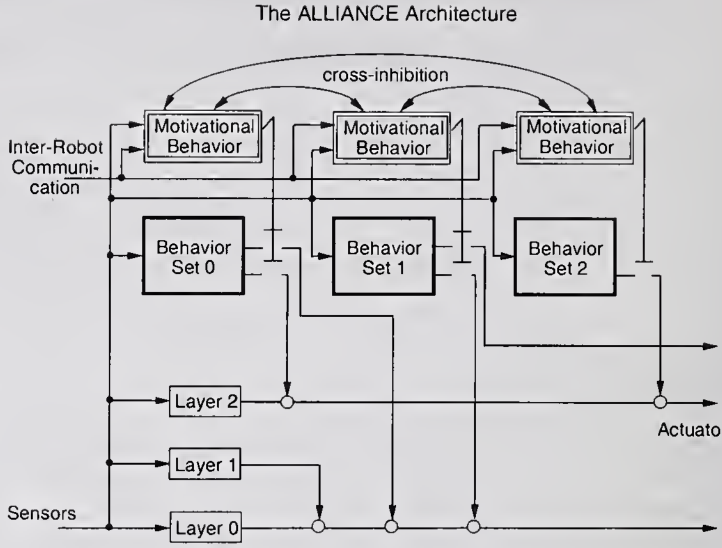


Figure 1: The ALLIANCE architecture for multi-robot cooperation.

The motivational behavior mechanism is based upon the use of two mathematically-modeled motivations within each robot – impatience and acquiescence – to achieve adaptive action selection. Using the current rates of impatience and acquiescence, as well as sensory feedback and knowledge of other team member activities, a motivational behavior computes a level of activation for its corresponding behavior set. Once the level of activation has crossed the threshold, the corresponding behavior set is activated and the robot has selected an action. The motivations of impatience and acquiescence allow robots to take over tasks from other team members (i.e., become impatient) if those team members do not demonstrate their ability – through their effect on the world – to accomplish those tasks. Similarly, they allow a robot to give up its own current task (i.e., acquiesce) if its sensory feedback indicates that adequate progress is not being made to accomplish that task.

We have shown that this approach can guarantee, under certain constraints, that the robot team will accomplish their objectives [15]. We have implemented this approach in a wide variety of applications in the laboratory on several different types of physical and simulated robot systems. Figures 2 and 3 illustrate these different implementations. The implementations include the "mock" hazardous waste cleanup [17], box pushing [20], janitorial service [16], bounding overwatch [16], formation-keeping [14], cooperative manipulation [18], cooperative tracking of multiple moving targets [19], and cooperative production dozing. These implementations and results now give us the basis for studying issues of metrics within this framework.

4 Evaluation of Metrics in ALLIANCE Applications

In [16], the ALLIANCE architecture was demonstrated to have the important qualities of robustness, fault tolerance, reliability, flexibility, adaptivity, and coherence, which we identified as critical design requirements for a cooperative multi-robot team architecture. These broad characteristics, however, were determined based upon qualitative evaluations of the various implementations we have performed. Ideally, we would prefer to have more quantitative metrics of evaluation for these higher-level team characteristics.

On a more application-specific level, we used several metrics to evaluate robot team performance within each of these applications. Table 1 summarizes the metrics we used to analyze the performance of multiple robot teams in eight different ALLIANCE implementations. In these applications, concrete indicators of mission success were used, such as numbers of objects moved, distance traveled, or number of targets within view. Improved mission quality was based upon the time taken to achieve these indicators. This is natural, since a primary benefit of multiple robot teams is using parallelism to achieve mission speedup. In these implementations, no single metric was found to be most useful. The need for a variety of metrics suggests that system performance measures are application-dependent. These examples also illustrate that, for typical applications, the most important issues are *whether* and *how well* the robot team completes its mission.

By focusing on application-specific metrics, however, the broader-perspective qualities of robustness, fault tolerance, adaptivity, etc., are not made explicit. Instead, these characteristics are hidden in the application-specific measures. Thus, any shortcomings in a robot team's ability to operate robustly or with a high degree of fault tolerance, for example, would be measured by an increased time to complete the mission (or by never completing the mission at all), a decreased distance traveled, fewer objects moved, etc. It would be difficult, therefore, to determine the relative levels of contribution of the various broader-perspective qualities (e.g., fault tolerance vs. adaptivity) to changes in the application-specific quantitative measures (e.g., distance traveled). Thus, if one wants to explicitly measure fault tolerance across several control architectures, and/or several application domains, these metrics are not suitable.

An important goal of research in the quantitative evaluation of robot control architectures is, therefore, the development of metrics that enable quantitative measurement higher-level characteristics, including fault tolerance, reliability, flexibility, adaptivity, and coherence. By averaging the results across multiple application domains, we would then be able to explicitly compare alternative control architectures in terms of these important application-independent characteristics. Our continuing research is

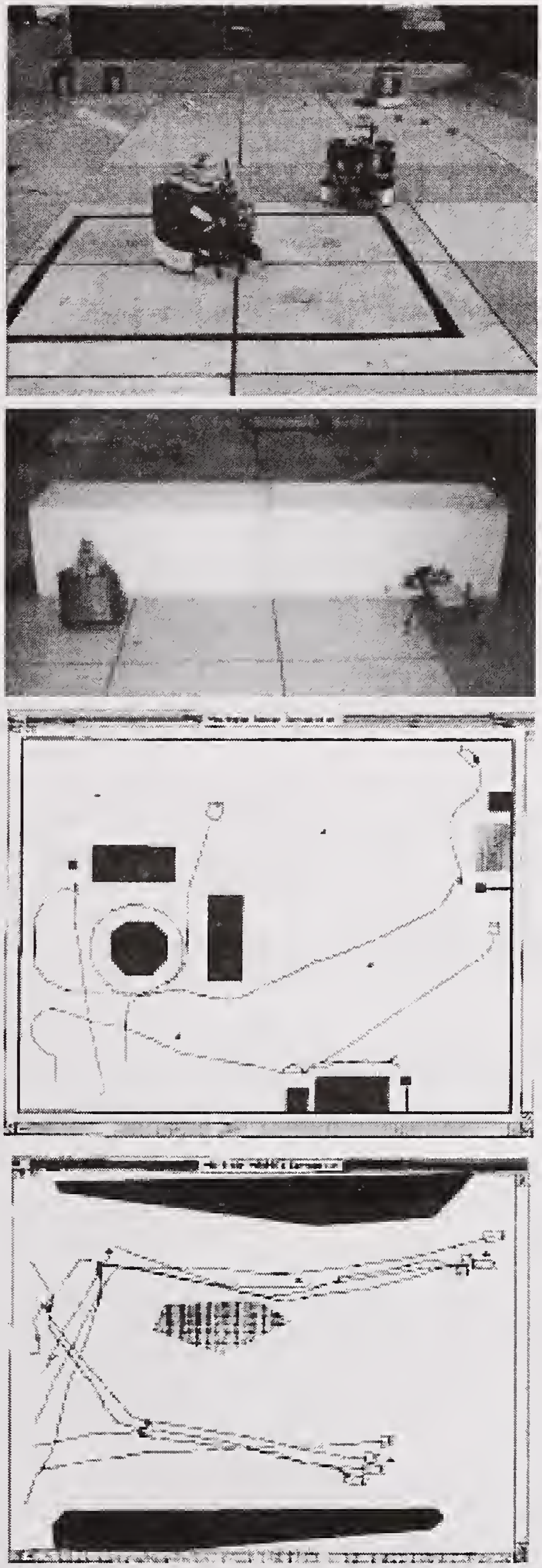


Figure 2: Implementations of the ALLIANCE architecture (on both simulated and physical robots). From top to bottom, these implementations are: "mock" hazardous waste cleanup, bounding overwatch, janitorial service, and box pushing.

Application domain	# Robots	Metric description	Metric definition
1. "Mock" hazardous waste cleanup	2-5 (P)	a. Time of task completion	t_{max}
		b. Total energy used	$\sum_{t=1}^{t_{max}} \sum_{i=1}^{m} e_i(t)$, where $e_i(t)$ is energy used by robot i through time t (m robots)
2. Box pushing	1-2 (P)	Perpendicular dist. pushed per unit time	$d_\perp(t)/t$, where $d_\perp(t)$ is $\perp$ distance moved through time t
3. Janitorial service	3-5 (S)	a. Time of task completion	t_{max}
		b. Total energy used	$\sum_{t=1}^{t_{max}} \sum_{i=1}^{m} e_i(t)$, where $e_i(t)$ is energy used by robot i through time t (m robots)
4. Bounding overwatch	4-20 (S)	Distance moved per unit time	$d(t)/t$, where $d(t)$ is distance moved through time t
5. Formation-keeping	4 (P & S)	Cumulative formation error	$\sum_{t=0}^{t_{max}} \sum_{i \neq leader} d_i(t)$, where d_i = distance robot i is misaligned at t
6. Simple multi-robot manipulation	2-4 (P)	Number of objects moved per unit time	$j(t)/t$, where $j(t)$ is number of objects at goal at time t
7. Cooperative tracking	2-4 (P) 2-20 (S)	Avg. number of targets observed (collectively)	$A = \sum_{t=1}^{t_{max}} \sum_{j=1}^{n} \frac{g(B(t),j)}{t_{max}}$, where $B(t) = [b_{ij}(t)]_{m \times n}$, ($m$ robots, n targets) $b_{ij}(t) = 1 \Longrightarrow$ robot i observing target j at t, $g(B(t),j) = \begin{cases} 1 & \text{if exists } i \text{ s.t. } b_{ij}(t) = 1 \\ 0 & \text{otherwise} \end{cases}$
8. Multi-vehicle production dozing	2-4 (S)	Quantity of earth moved per unit time	$q(t)/t$, where $q(t)$ is quantity of earth moved through t

Table 1: Summary of metrics used in ALLIANCE implementations. (In the second column, "P" refers to physical robot implementations; "S" refers to simulated robot implementations.)

aimed at developing these higher-level metrics for the evaluation of robot team performance.

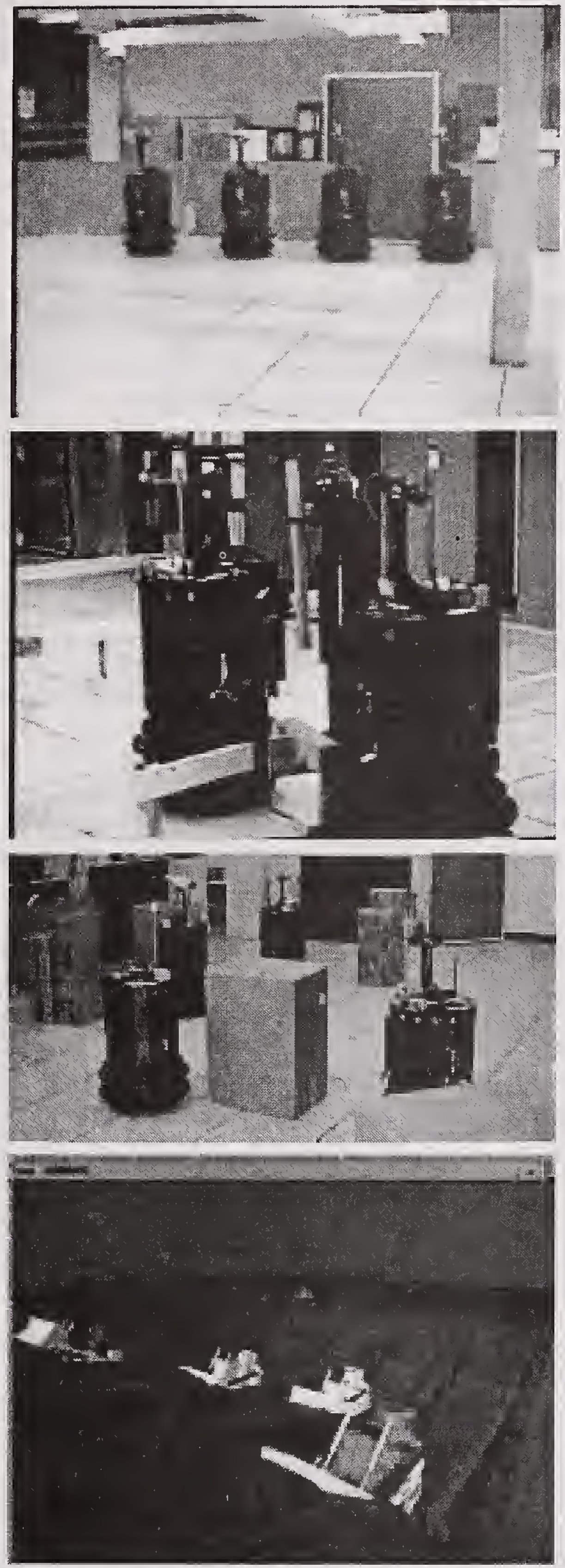


Figure 3: Additional implementations of the ALLIANCE architecture. From top to bottom, these implementations are: cooperative manipulation, formation-keeping, cooperative tracking of multiple moving targets, and cooperative production dozing.

Acknowledgments

This article has been authored by a contractor of the U. S. Government under Contract No. DE-AC05-00OR22725. Accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U. S. Government purposes. Research sponsored in part by the Engineering Research Program of the Office of Basic Energy Sciences, U. S. Department of Energy. Oak Ridge National Laboratory is managed by UT-Battelle, LLC for the U.S. Dept. of Energy under contract DE-AC05-00OR22725.

References

[1] R. Alami, S. Fleury, M. Herrb, F. Ingrand, and F. Robert. Multi-robot cooperation in the Martha project. *IEEE Robotics and Automation Magazine*, 1997.

[2] T. Arai, H. Ogata, and T. Suzuki. Collision avoidance among multiple robots using virtual impedance. In *Proceedings of the Intelligent Robots and Systems (IROS)*, pages 479–485, 1989.

[3] T. Balch. Social entropy: An information theoretic measure of robot team diversity. *Autonomous Robots*, 8(3), 2000.

[4] T. Balch and R. Arkin. Behavior-based formation control for multi-robot teams. *IEEE Transactions on Robotics and Automation*, December 1998.

[5] Tucker Balch and Ronald C. Arkin. Communication in reactive multiagent robotic systems. *Autonomous Robots*, 1(1):1–25, 1994.

[6] J. Deneubourg, S. Goss, G. Sandini, F. Ferrari, and P. Dario. Self-organizing collection and transport of objects in unpredictable environments. In *Japan-U.S.A. Symposium on Flexible Automation*, pages 1093–1098, Kyoto, Japan, 1990.

[7] Alexis Drogoul and Jacques Ferber. From Tom Thumb to the Dockers: Some experiments with foraging robots. In *Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, pages 451–459, Honolulu, Hawaii, 1992.

[8] Douglas Gage. Randomized search strategies with imperfect sensors. In *Proceedings of SPIE Mobile Robots VIII*, Boston, September 1993.

[9] David Jung and Alexander Zelinsky. Grounded symbolic communication between heterogeneous cooperating robots. *Autonomous Robots*, 8(3), July 2000.

[10] S. Mahadevan and J. Connell. Automatic programming of behavior-based robots using reinforcement learning. In *Proceedings of AAAI-91*, pages 8–14, 1991.

[11] S. Marsella, J. Adibi, Y. Al-Onaizan, G. Kaminka, I. Muslea, and M. Tambe. On being a teammate: Experiences acquired in the design of RoboCup teams. In O. Etzioni, J. Muller, and J. Bradshaw, editors, *Proceedings of the Third Annual Conference on Autonomous Agents*, pages 221–227, 1999.

[12] Maja Mataric. *Interaction and Intelligent Behavior.* PhD thesis, Massachusetts Institute of Technology, 1994.

[13] David McFarland. Towards robot cooperation. In D. Cliff, P. Husbands, J.-A. Meyer, and S. Wilson, editors, *Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, pages 440–444. MIT Press, 1994.

[14] L. E. Parker. Designing control laws for cooperative agent teams. In *Proceedings of the IEEE Robotics and Automation Conference*, pages 582–587, Atlanta, GA, 1993.

[15] L. E. Parker. *Heterogeneous Multi-Robot Cooperation.* PhD thesis, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Cambridge, MA, February 1994. MIT-AI-TR 1465 (1994).

[16] L. E. Parker. On the design of behavior-based multi-robot teams. *Journal of Advanced Robotics*, 1996.

[17] L. E. Parker. ALLIANCE: An architecture for fault-tolerant multi-robot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220–240, 1998.

[18] L. E. Parker. Distributed control of multi-robot teams: Cooperative baton-passing task. In *Proceedings of the 4th International Conference on Information Systems Analysis and Synthesis (ISAS '98)*, volume 3, pages 89–94, 1998.

[19] L. E. Parker. Cooperative robotics for multi-target observation. *Intelligent Automation and Soft Computing, special issue on Robotics Research at Oak Ridge National Laboratory*, 5(1):5–19, 1999.

[20] L. E. Parker. Lifelong adaptation in heterogeneous teams: Response to continual variation in individual robot performance. *Autonomous Robots*, 8(3), July 2000.

[21] N. S. V. Rao. Terrain model acquisition by mobile robot teams and n-connectivity. In *Proceedings of the Fifth International Symposium on Distributed Autonomous Robotic Systems (DARS 2000)*, 2000.

[22] I. Rekleitis, G. Dudek, and E. Milios. Graph-based exploration using multiple robots. In *Proceedings of the Fifth International Symposium on Distributed Autonomous Robotic Systems (DARS 2000)*, 2000.

[23] D. Rus, B. Donald, and J. Jennings. Moving furniture with teams of autonomous robots. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 235–242, 1995.

[24] D. Rus and M. Vona. A physical implementation of the self-reconfiguring crystalline robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1726–1733, 2000.

[25] D. Stilwell and J. Bay. Toward the development of a material transport system using swarms of ant-like robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 766–771, Atlanta, GA, 1993.

[26] C. Unsal and P. K. Khosla. Mechatronic design of a modular self-reconfiguring robotic system. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1742–1747, 2000.

[27] A. Yamashita, M. Fukuchi, J. Ota, T. Arai, and H. Asama. Motion planning for cooperative transportation of a large object by multiple mobile robots in a 3d environment. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 3144–3151, 2000.

[28] M. Yim, D. G. Duff, and K. D. Roufas. Polybot: a modular reconfigurable robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 514–520, 2000.

Shared Autonomy and Teaming: A preliminary report[*]

Henry Hexmoor[αβ] and Harry Duchscherer[γ]

[α] Computer Science & Computer
Engineering Department, Engineering Hall,
Room 313, Fayetteville, AR 72701
[β]Center for Multisource Information Fusion
University at Buffalo, Buffalo, NY 14260

[γ]University of North Dakota
Grand Forks, North Dakota, 58202

ABSTRACT

We outline how an agent's shared autonomy considerations affect its interaction in a team. A unified model of acting and speaking will be presented that includes teaming and autonomy. This model is applied to the domain of satellite constellation. We introduce our simulator and outline our application of autonomy and teaming concepts.

KEYWORDS: Multi-agent Systems, Shared Autonomy, Agent Teams

1. INTRODUCTION

We have presented Situated Autonomy as a moment-by-moment attitude of an agent toward a goal and have argued that it is a useful notion in modeling social agents [6].

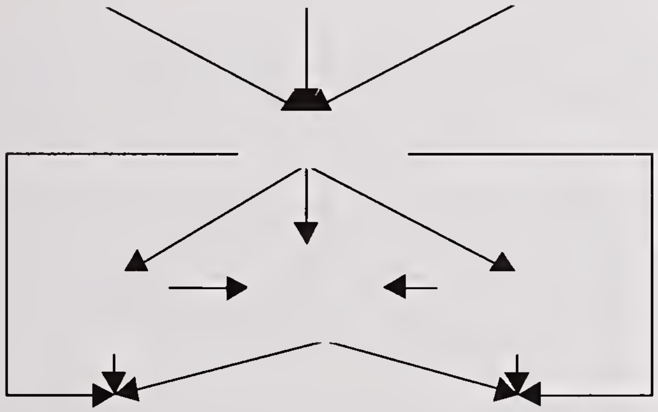


Figure 1 Action Selection

We argued that a combination of the nature and the strength of an agent's beliefs and motivations lead the agent to perceive one of the following: (a) the agent chooses itself to be the executor of the goal, (b) the agent delegates the goal entirely to others, (c) the agent shares its autonomy with other agents, or (d) the agent has a relatively small and undetermined responsibility toward the goal. Our focus in this paper is when the agent perceives shared autonomy.

Situated autonomy is an important part of an agent's action selection. Figure 1 shows a very simple action selection in Belief Desire Intention (BDI) paradigm and the role of situated autonomy. Along with goals and beliefs, we believe situated autonomy is used in the process of determining intentions. The process can be highly cognitive as in planning or less cognitive as in reaction generation. Enablers are the agent's perception of its own abilities, social factors, tools, and resources.

There are many accounts of starting or joining a team [2, 4, 10]. We favor the ingredients of intentional cooperation laid out by Tuomela: (a) collective goal or plan, (b) strong correlation among member's interest or preferences, and (c) having a cooperating and helping attitude.

We believe that in common situations, an agent's situated autonomy changes at a lot faster pace than its participation in a team. Once an agent perceives shared autonomy toward a goal, it may be inclined to recruit one or more agents to form a team. After a team is formed, the agent's degree of shared autonomy will change at the speed of perceived changes to the cognitive ingredients of situated autonomy. A recruited agent's degree of shared autonomy will be smaller than the recruiter's shared autonomy but after a team is formed will change to any level.

We are developing a model that unifies acting and speaking [7]. This model uses production rules to encode a conversational policy. A conversational policy is a modeling system that is designed to encode a set of conventions shared among a group of agents [5]. Such systems are generally called Normatives [1,10]. A prototypical agent follows the conventions of the group in communicating and sharing mental states. However, situated autonomies of each agent will individualize its interactions and allow it to deviate from the expected behavior.

[*] This work is supported by AFOSR grant F49620-00-1-0302.

We will present a model of conversational model in generic, non-BDI format. Each agent will personalize parts of conversational policy in its own BDI paradigm. A conversation policy is two types of simple production-like structures we will call transitions, shown below.

```
physical condition * spoken word/phrase * speak state
                ➔ speak state
        physical condition * speak state
             ➔ spoken word/phrase
```

To model physical actions of an agent in reactive behaviors, we introduce two other types transitions, shown below.

```
physical condition * spoken word/phrase * act stat
                ➔ act state
        physical condition * act state
                  ➔ act
```

A number of agents may share a unified model. For example, a group of agents may share a conversational policy. The shared model becomes their Norm. By entering a model and tracking the shared states, agents can synchronize their actions. Privately, each agent will consider transitions in terms of beliefs or goals, and intentions.

In the general model, physical conditions arbitrate among productions that provide alternative acts or words at a given state. However, each agent will have a personalized perception and interpretation of the physical conditions in terms of beliefs. We consider agents' situated autonomy and teaming consideration is determined by the agent's unique perceptions of the common physical conditions. Below we rewrite the transitions from an agent's perspective and add situated autonomy. 'Physical conditions' and 'spoken word/phrases it hears' are things about which an agent has beliefs. The states are the agent goals (or interchangeably desires). The 'physical act' or 'chosen word/phrase for communication' are the objects of an agent's intentions.

```
  Belief(physical condition) *
 Belief(spoken word/phrase) *
        Goal(speak)
       ➔ Goal(speak)
  Belief(physical condition) *
       Goal(speak) *
     situated autonomy
➔ Intention(spoken word/phrase)
  Belief(physical condition) *
 Belief(spoken word/phrase) *
         Goal(act)
        ➔ goal(act)
  Belief(physical condition) *
        Goal(act)  *
     situated autonomy
      ➔ Intention(act)
```

The remainder of this paper is organized by working through an example of a unified model and how agents can personalize the physical conditions and consider teaming and changes in their Shared Autonomy. We will present a simulation of a constellation of satellites that can be tasked from ground. We will show our unified model and related issues of learning autonomy level using this application domain. We have not yet conducted experiments with situated autonomy and hence we consider this report a preliminary report.

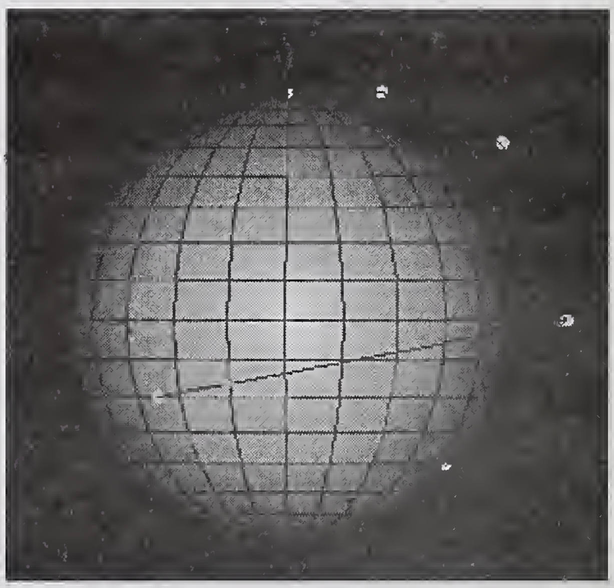


Figure 2. The Server's graphic screen

2. SIMULATION OF A CONSTELLATION OF SATELLITES

We have developed our own satellite simulator to illustrate our research ideas outlined in this paper. The simulator follows the principles of TechSat 21 [8]. SaVi is a similar software created at the Geometry Center at the University of Minnesota for the visualization and analysis of satellite constellations [3]. It has been used to simulate various satellite constellations such as Globalstar, Iridium, and Teledesic. SaVi differs from ours in that it is simply a simulator of orbital satellite constellations, and does not implement autonomy in its satellites.

Our simulator is composed of two primary modules; the server, and the agent. The server module handles the creation of all agent objects in the simulation and acts as a router to facilitate the passing of messages between various agents. There are two types of agents that can be created within this environment, satellite agents and ground station agents. These agents are implemented as objects and have similar capabilities, with the satellites having the additional ability to change their location within the environment. The server module is also responsible for the accurate representation of all objects in the graphical environment (Figure 2).

The agent module contains the functional components of the agents. These components constitute the essence of the agent's purpose and functionality. Behavioral functions and autonomy states can be created and transitioned by accessing these module components through the use of behavior rules in the agent's behavior file. Behavior rules are comprised of conditional checks and assignment calls to the functional components in the form of simple production rules.

The satellite simulator was implemented using Mesa and supported by the collision detection routines which are part of the SOLID library package. The simulation is comprised of a central solid sphere surrounded by a wire-frame sphere to establish a latitude/longitude coordinate system. The sphere is currently scaled to represent the earth, and rotational velocity is approximately 120 times nominal. Graphically, the satellites are represented as green spheres with groundstations being yellow spheres on the planet's surface. The entire simulation can be rotated on any of the three axises. This allows for the simulation to be viewed from various prespectives. Additionally, any agent can be selected to be "tracked" in the simulation. This has the affect of centering the agent at the origin, with all other objects, including the planet, revolving around the agent. Blue line segments are used to show connections between satellites that have a line of sight communications capability, or connections between a satellite and a ground station (Figure 4). The SOLID library was used to make this determination, since the Mesa libraries do not directly support the detection of intersections between the connecting line segments and the planetary bodies. All satellites orbit at velocities which are appropriate for their altitude, with respect to a planet such as the earth.

The satellites and ground stations that orbit and reside on the planet are implemented as objects and have communication capability to other agents via message passing through the socket connection with the server. The position of each of these agents is determined by the data that is provided to the server in a text file. The text file contains only the most basic of information necessary to place the agent in the graphics environment of the server.

As each agent object is created, it reads a behavior file, which contains the rules that will govern its actions with respect to communication policy and physical actions that may be needed to achieve a desired goal. The format and examples of these rules is described in more detail in the next section.

3. TAKING 3 SCANS OF AN AREA

Assume the ground station will need three independent images of a given longitude and latitude from a given altitude. Let's call the task 3Image. The ground station issues the command to the nearest satellite and that satellite will be responsible to perform the task either by itself if no satellites are available. The satellite will complete the images itself taking one image in each orbit crossing the given location. If the satellite so decides it recruits other satellites to complete the task. Each of the recruited satellites may recruit another satellite. After recruiting one satellite, either satellite may decide to recruit a third teammate.

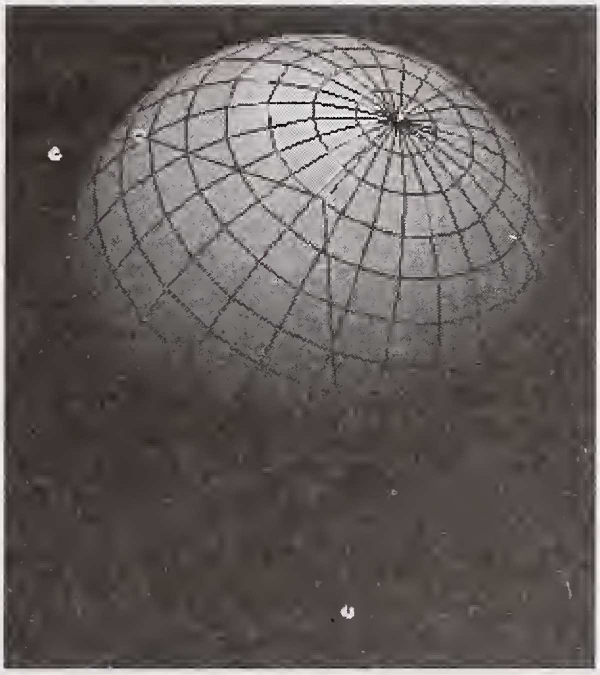


Figure 4. Communication lines

Here we will present a conversational policy that will govern interagent communication.

The following are the agent *speak states*:
0 – Start state
1 – Ground station has issued a command and a Satellite has received this message.
2 – A satellite has received and accepted the command.
3 – A second satellite has been contacted.
4 – The second satellite has accepted the command.
5– A third satellite has been contacted.
6– The third satellite has accepted the command and we now have a team.
7- Ground control has received the first image.
8- Ground control has received the second image.
9- Ground control has received the third image.
10- Success State

11- Failure State. This state occurs when any of the images are not received in a reasonable amount of time. State 0 is the start of 3-imaging.

The following are the set of available *words/phrases*:

S0 – Satellite agent says “Hello” to other agents to announce its presence, if it is currently idle.
S1 – Ground station issues a command 3Image [Longitude] [Latitude] [Altitude]
S2 – A satellite accepts command. The satellite says "Roger to 3Image"
S3 – Ground states acknowledges that a team leader has agreed to take the task and will now accept images by speaking “Ready to receive images”.
S4 – A satellite recruits another satellites for 3Image. The satellite may say “Team 3Image?”
S5 – If a satellite accepts the request for being part of a team for 3Image, it may say “Willco”.
S6 - If a satellite rejects the request for being part of a team for 3Image, it may say "Unable".
S7 – “bye” is spoken when a team member is no longer able to be part of the team.
S8 – “Downloading Image #1” is spoken when image #1 is downloaded to the ground Station.
S9 – “Downloading Image #2” is spoken when image #2 is downloaded to the ground Station.
S10 – “Downloading Image #3” is spoken when image #3 is downloaded to the ground Station.
S11 – “Received Image #1” is spoken when image #1 is received by the ground Station.
S12 – “Received Image #2” is spoken when image #2 is received by the ground Station.
S13- The ground station may say “Task Complete” when all three images are received.
S14- With an excessive silence, the policy ends unsuccessfully, “Task Aborted”.

The following are the *physical conditions.* For each condition we note the agent that perceives it.

P0 – Start condition.
P1 - There is a need for 3Imgaing and a satellite is chosen for tasking. This condition is perceived by GROUND only.
P2 - Satellite is unable to participate in a team for one of two reasons: It is in danger or it has not yet finished its previous task. This condition is perceived by the SAT that is contacted to perform the task.
P3 – Satellite is able to take lead on a task and is available. This condition is perceived by SAT only.
P4 – Another satellite is detected that can potentially be a team-mate. This condition is perceived by SAT.
P5- Satellite is able to be a team-player. This condition is perceived privately by the SAT. All SAT agents privately perceive conditions P6-P10.
P6- An image has been collected.
P7- An image has been successfully collected and transmitted to the ground station.
P8- Two images are successfully collected and transmitted to the ground.
P9- Three images are successfully collected and transmitted to the ground.
P10– The chosen Satellite has received the command. Ground station is now ready to receive images. This condition is perceived by GROUND only.
P11- All the external conditions and instrumentation conditions for taking a picture are met.

In the following *speak state transitions,* each agent’s type is noted by a “GND” for ground station or “SAT” for satellite. SATAVL, TIMEOUT, UNABLE, AND PICT are boolean conditions. SATAVL determines if a satellite agent is available (free of prior tasks and capable of taking on new a task) for the current agent. The availability is determined with respect to the satellite’s current speak state and physical conditions. TIMEOUT holds if an excessive amount of time has elapsed since the last change in speak state. PICT indicates if the agent has any pictures that can be downloaded to the ground station. UNABLE denotes the satellite’s propioception of being busy with a prior task or somehow being “out of service”. PICT denotes the absence of such a condition. “SPK:<*destination*>” construct is used to specify to whom the spoken phrase is intended. CUR_AGNT is the agent most recently identified as available by the SATAVL check. The default CUR_AGNT is the speaking agent.

The following are the *speak-state transitions.*

```
P0*1*GND*S1➔0

P3*0*SAT*S1➔1
P3*1*SAT*S3➔2
P4*2*SAT*S4➔3
P4*4*SAT*S4➔5
P4*3*SAT*S6➔2
P4*3*SAT*S4➔4
P4*5*SAT*S5➔6
P5*0*SAT*S3➔2
P2*3*SAT*S7➔0
P2*4*SAT*S7➔0
P2*5*SAT*S7➔0
P2*6*SAT*S7➔0
P2*7*SAT*S7➔0
P2*8*SAT*S7➔0
P4*4*SAT*S7➔2
P4*5*SAT*S7➔3
P4*6*SAT*S7➔4
P7*2*SAT*S11➔7
P7*4*SAT*S11➔7
P7*6*SAT*S11➔7
P6*2*SAT*S11➔7
P6*4*SAT*S11➔7
P6*6*SAT*S11➔7
P8*7*SAT*S12➔8
```

```
P6*7*SAT*S12→8
P7*7*SAT*S12→8
P9*8*SAT*S13→9
P6*8*SAT*S13→9
P7*8*SAT*S13→9
P8*8*SAT*S13→9
P10*0*GND*S2→1
P10*1*GND*S3→2
P10*2*GND*S8→7
P10*7*GND*S9→8
P10*8*GND*S10→9
P10*9*GND*S13→10
1*SAT*S14→11
2*SAT*S14→11
3*SAT*S14→11
4*SAT*S14→11
5*SAT*S14→11
6*SAT*S14→11
7*SAT*S14→11
8*SAT*S14→11
P10*1*GND*TIMEOUT→11
P10*7*GND*TIMEOUT→11
P10*8*GND*TIMEOUT→11
```

The following are the *speak transitions.* SA denotes the agent's level of situated autonomy.

```
P0*0*SAT*TIMEOUT→SPK:ALL*S0
P1*0*GND*TIMEOUT→SPK:CUR_AGNT*S1
P0*0*SAT*S4→P5
P3*1*SAT→SPK:ALL*S2
P10*1*GND→SPK:ALL*S3
P4*2*SAT*SA→SPK:ALL*S4
P5*0*SAT*S4→P2
P6*2*SAT*SA→SPK:ALL*S5
P7*2*SAT*SA→SPK:ALL*S8
P4*4*SAT*SA→SPK:ALL*S4
P2*4*SAT*SA→SPK:ALL*S7
P7*4*SAT*SA→SPK:ALL*S8
P2*6*SAT*SA→SPK:ALL*S7
P7*6*SAT*SA→SPK:ALL*S8
P2*0*SAT→SPK:ALL*S6
P2*3*SAT→SPK:ALL*S7
P2*5*SAT→SPK:ALL*S7
P2*7*SAT→SPK:ALL*S7
P2*8*SAT→SPK:ALL*S7
P8*7*SAT→SPK:ALL*S9
P9*8*SAT→SPK:ALL*S10
P10*7*GND→SPK:ALL*S11
P10*8*GND→SPK:ALL*S12
P10*9*GND→SPK:ALL*S13
P10*11*GND→SPK:ALL*S14
```

The following are the *act transitions.* "A" denotes an act, which in 3Imaging is taking a picture.

```
P11*2*SAT*SA→A
P11*4*SAT*SA→A
P11*6*SAT*SA→A
```

In addition to the conversational policy and action rules (above), we have designed rules for our agents to infer physical conditions based on exiting physical conditions and their current speak states and either (a) what they hear, (b) propioception of time or success of their own task (taking a picture), or (c) perception (availability of another satellite for teaming). We will consider these rules to be more domain oriented and intended for internal use of agents. Collectively, we will refer to these rules as domain rules.

The following are mainly based on hearing.

```
P1*0*GND*S2→P10
P2*0*SAT*S7→P0
P0*0*SAT*S1→P3
P4*4*SAT*S5→P3
P0*0*SAT*S3→P5
```

The following are mainly based on agent perception.

```
P0 * 0 * GND*SATAVL→ P1
P3*2*SAT*SATAVL→P4
P3*4*SAT*SATAVL→P4
```

The following are mainly based on agent propioception.

```
P1*1*GND*S1*TIMEOUT→P0
P5*2*SAT*PICT→P6
P4*6*SAT*PICT→P6
P4*6*SAT*PICT→P7
P6*2*SAT*PICT→P7
P6*4*SAT*PICT→P7
P6*7*SAT*PICT→P8
P7*7*SAT*PICT→P8
P6*8*SAT*PICT→P9
P7*8*SAT*PICT→P9
P8*8*SAT*PICT→P9
UNABLE →P2
```

4. USING CONVERSATIONAL POLICY

Agents can use the conversational policy for forming their beliefs, goals, and intentions. Each agent will apply the policy, action, and domain rules to new messages it receives. The following is our highest-level loop pseudo code for agent update.

```
For (agent; 1; numAgents)
  While (new receive message)
  {
    1. Determine SA
    2. For (rule; 1; numRules)
         If (rule applies)
           a. Perform transitions
              Use SA to resolve conflicts
           b. Update beliefs and goals
    3. Perform the intention for speaking or acting
       within reaction constant
  }
```

Given a goal and the prevailing physical conditions agents constantly update their SA. SA is used in resolving conflicts in rules and in final decision of intention to be formed. Based on situated autonomy agents perform their picture taking or recruit other agents as teammates. The GND agent will note P0 or P1 (and form a belief) and will instantiate an instance of 3Imaging conversational policy. GND will maintain state 0 as its goal. Being in state 0 and having perceived P1, GND will use a speak transition to intend and then to issue S1. If the satellite (call it SAT1) has received the message S1 the speak state transition is used to reach state 1. GND and satellite SAT1 share the goal of being in state 1. SAT1 may perceive P3 and using a speak state transition to arrive at a desire to be in state 2 and also form an internal goal in achieving the command. GND does not determine P3 so it has no access to this perception. It however has access to the state transition that allows it to desire state 2. In state 2, SAT1 privately considers P3, P4, and P11 and arrives at a determination of situated autonomy. In 3Imaging, the lead agent once it reaches state 2, must consider exogenous physical conditions 3, 4, and 11 along with all agent endogenous factors to determine its autonomy. If it decides on shared autonomy, the agent must begin recruiting other agents as teammates. Otherwise, it will either do the task itself or delegate it to others.

If SAT1's decision favors a team formation, it uses a state transition to arrive at state 3 and forms a desire in it. Due to space limitation, we will not discuss the details of team formation. Since P4 is not shared with GND, it does not have the same belief. Let's call the second Satellite SAT2. SAT1 and SAT2 now share the desire to be in state 3. If SAT2 perceives P5, it will use a state transition and moves to state 4 and forms a desire in state 4 and the goal to be a teammate in 3Imaging. If SAT2 perceives P2, it will inform SAT1 and move back to state 2. SAT2 no longer has to want state 3. SAT1 will desire State 2.

For an agent that is recruited to be a teammate in state 4 it has already decided to have shard autonomy. It must consider its exogenous physical conditions 3 and 4 (P3 and P4) along with all agent endogenous factors to determine its autonomy in order to decide whether yet another teammate is needed. If it decides to recruit another agent it will move through states to state 6.

For an agent that is recruited to be a teammate in state 6 it has already agreed to have shard autonomy and since it is the third member of the team no other teammates are needed. Conditions P6-P9 may be perceive by either Satellite agent and all SAT agents share goals in state 7-11.

In the next section we will briefly discuss how autonomy will vary.

5. AUTONOMY MEASURES

Situated Autonomy depends on time, and strengths of belief and goal. [6]. Each agent reacts at different speeds. The times between sensing and acting is an agent's reaction constant and the optimal values can be learned. This greatly affects the agent's autonomy decision. Temporally, from the shortest reaction time to the longest, an agent's autonomy is based on it's pre-disposition, disposition, and motivation. Therefore, an agent's reaction constant is important. An agent's beliefs used in autonomy decision vary from weak to strong. An agent's goals are directed to self, other, or group. The goals vary in strength of motivation from weak to strong.

In 3Imaging, agents have different reaction constants and we are experimenting with the effect of slow versus fast reacting satellites. An agent's beliefs are about the physical conditions and the speak states and change in strength. The goals are about taking images and they vary based on the agent's prior commitment. If a Satellite agent has committed to a 3Imaging task, it might commit to yet another 3Imaging command if it senses that it can complete the task. After the first command, the motivation level for the goal is set to be less than for the first command. A combination of belief and goal degrees are uscd for determining SA.

As of this writing, our implemented system runs and images are gathered. However, we do not yet have situated autonomy experiments. We plan to compare runs of the system with different reaction constants. The autonomy levels in our agents will be learned as combinations of beliefs and goals. The metrics we will use for feedback are timeliness of images collected.

6. SUMMARY AND CONCLUSION

We have developed a production-style representational framework that unifies acting and speaking. Our representation extends conversational policy scheme. It explains how agents can use the shared normative models of conversational policy for forming private beliefs, goals, and intentions. We outlined a scheme for flexible teaming that uses the notion of situated autonomy.

We have implemented our model in the domain of constellation of satellites. Our system runs but we have not yet completed experiments with how timely team formation improves our system performance.

REFERENCES

[1] C.E. Alchourron and E. Bulygin, (1971). Normative systems, Springer Verlag, Wien.

[2] P. Cohen, H. Levesque, 1. Smith, (1997), On Team Formation, In J. Hintika and R. Tuomela, **Contemporary Action Theory**, Synthese.

[3] G. Bergen (1998), **SOLID - Interference Detection Library,** Department of Mathematics and Computing Science, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands. (http://www.win.tue.nl/cs/tt/gino/solid/solid2_toc.html)

[4] F. Dignum, B. Dunin-Keplicz, and R. Verbrugge, (2000), Agent Theory for Team Formation by Dialogue, In Proceedings of ATAL-2000, Boston.

[5] M. Greaves, H. Holmback, and J. Bradshaw, (1999). *What is Conversation Policy?* In Autonomous Agents (Agents-99) Workshop titled Specifying and Implementing Conversation Policies, Seattle, WA.

[6] H. Hexmoor, (2000a). A Cognitive Model of Situated Autonomy, In Proceedings of **PRICAI-2000** Workshop on Teams with Adjustable Autonomy, Australia.

[7] H. Hexmoor, (2000b). Conversational Policy: A case study in air traffic control, In Proceedings of International Conference in Al, **IC-AI-2000**, Los Vegas.

[8] TechSat 21: Advanced Research and Technology Enabling Distributed Satellite Systems, *Overview Briefing of TecdhSat 21*, http://www.vs.afrl.af.mil/vsd/techsat21.

[9] R. Tuomela, (2000), **Cooperation**, Kluwer Pub.

[10] A. Valente and J. Breuker, (1994). A Commonsense Formalization of Normative Systems, In Proceedings of the ECAI-94 Workshop on Artificial Normative Reasoning, J. Breuker (ed), Amsterdam, p. 56-67.

Real Time Distributed Expert System for Automated Monitoring of Key Monitors in Hubble Space Telescope

Reza Fakory, Ph.D.
Computer Sciences Corporation
mfakory@csc.com

Majid Jahangiri, Ph.D.
Computer Sciences Corporation
mjahangiri@v2pop.hst.nasa.gov

Abstract

A distributed expert system for monitoring the critical telemetry (the Key Monitors) of Hubble Space Telescope (HST) has been designed and developed. The Key Monitors Expert System (KMES) monitors the general health of the space craft operation through analysis of the Key Monitors data. KMES uses rule-based approaches and notifies operators/system engineers when it receives a limit violation from Front End Processor subsystem (FEP). The design of KMES is similar to the design of a previously reported system called "Expert System for Automated Monitoring" (ESAM) which was developed for HST [1]. However, KMES uses an approach different from ESAM's approach. ESAM was designed to monitor all telemetry mnemonics in a selected subsystem via establishment of tight limits for mnemonics. On the other hand, KMES has been designed to monitor the Key Monitors, providing notifications for out-of-limit conditions in accordance with documented operational procedures. Upon detection of an out of limit conditions, KMES analyzes data for contingencies. It fires appropriate rules to request associated engineering data from telemetry repositories. Subsequently, KMES sends e-mails and e-pages to notify the appropriate System Engineers (SEs) and Operators. The duration of a limit violation is monitored to eliminate transient faults. KMES logs all out of bound (limits) violations but only takes an action for each persistent violation. In addition, the distributed system design approach of KMES allows a pre screening of data variations to reduce the number of queued rules. Also, design of KMES was modified to include only selected part (sub-database) of a main database into KMES's subprocesses. The sub-database contains data associated with mnemonics that are used within the associated subprocess. This approach significantly reduced the required real-time execution time and the memory usage for the expert system.

KMES also allows the user to override any activated miscompare. This feature permits operators to adjust for known anomalies or changes in operational context. The system generates event messages to override actions; the events include a user login ID and the reason for the override.

Currently, KMES includes rules to monitor seven subsystems. KMES rules can be expanded to include rules for other subsystems. This paper describes the fundamental design and features of KMES. The results for a simulated scenario leading to failure of a Key Monitor and timely detection of the failure by KMES and ESAM are presented.

1 Background

The Vision 2000 Command and Control System (CCS) Product Development Team has been formed to reengineer the HST ground system [2,3]. The CCS ground system consists of several systems including System Monitoring & Analysis (SYM). Development of an expert system for telemetry monitoring, fault detection and recovery for the HST is one of the SYM's responsibilities.

Prior to design of KMES, the SYM group developed a real-time Expert System for Automated Monitoring (ESAM) [1]. The system was designed and developed to monitor the general health of the spacecraft and to detect faults within the Hubble Space Telescope (HST) via monitoring all telemetry mnemonics within a selected subsystem. It employs model-based/rule-based, hierarchical fault tree analysis with forward-chaining rule propagation to compare expected state values with true states. The system uses a custom-built neural network model and System Engineer (SE)-provided algorithms to dynamically derive the expected state values based on knowledge of real-time or stored spacecraft commands. During operations, real-time telemetry values (i.e., true states) are compared to the expected state values for possible limit violations. The duration of a limit violation is monitored to eliminate transient faults. The system logs all miscomparisons but only issues a system event message for each persistent miscomparison. The persistence implementation approach significantly reduces the number of false miscompare messages. Currently, ESAM only includes rules and models associated with fault detection in Electrical Power System (EPS) of HST. Further expansion of ESAM for monitoring other subsystems of the spacecraft encountered two problems. First, for acquisition of telemetry data, from Information Sharing Protocol (ISP) into the expert system, a shared memory technique was employed to overcome synchronization between RTserver, the expert system server [1], and the ISP server. This design employed RTdaq, a COTS product from Talarian Inc. [6], that acquired data from shared memory and transferred data to RTserver. Further tests and analysis of results revealed that occasionally data was dropped during transmission from the shared memory to RTdaq. In addition, RTdaq did not have provisions for transmitting status flags that accompany the telemetry data from ISP, which indicate the general health and validation of the data. Second, modeling and development of rules, for incorporation of dynamic limits, required a significant amount of time from experts and system engineers with high level of expertise in the relevant subsystems of HST.

In order to overcome the first problem, it was decided to develop new modules with direct interface between RTserver and ISP via an existing middleware. For the second problem, it was decided to monitor the critical telemetry (the Key Monitors) and notify experts in accordance with Key Monitors documentation [7]. In this design, the limit values are constants that are defined in the Project Reference Database (PRD). The Front End Processor (FEP) subsystem of CCS detects limit violations for all monitors. KMES receives the Key Monitor values as well as the companion status flags from FEP. The status flag indicates limit violated Key Monitors. These new enhancements were incorporated into the design of KMES, and delivered as a part of a CCS Release delivery. The following sections describe the design features of the developed system.

2 Introduction

Expert systems are corner stones of knowledge Management [4,5] foundations and as such are designed to reduce dependency on humans and increase reliability of complex systems. For many cases, expert systems are simply a way to codify the explicit and sometimes tacit knowledge of experts (operators and system engineers) so it can be used to provide guidance and solutions for known problems. The real time Key Monitors Expert System (KMES) was designed and developed to automate the experts monitoring of the Key Monitors. In concept, KMES has been developed to automatically monitor the general health of spacecraft operation and notify operators and system engineers upon recognition of defined anomalies.

The Key Monitors are defined in the HST Contingency Plan document [7]. This document establishes a consistent and approved response to out of limit conditions or misconfigurations throughout mission operations. The out-of-bound limits have constant values defined in the Project Reference Database (PRD). In general, PRD includes two sets of limits namely yellow limit and red limit. For some Key Monitors mnemonics, the yellow and red limits coincide. In these cases, the response associated with red limit violation has priority over the response associated with the yellow limit violation. The FEP subsystem determines violated telemetry and sets a status flag, which accompanies the mnemonic value. For example, the FEP sets the companion status flag for a mnemonic to "L" when the telemetry value drops below the lower value of the red limit associated with the mnemonic. The following sections describe the developed system.

3 System Description

KMES is primarily a rule-based expert system. KMES subscribes and receives the Key Monitors mnemonic values and the companion status flags from ISP. Most of KMES rules are simple and the hierarchy is shallow. However, the required actions for some limit violations are contingent on configuration or statuses of other equipment. Therefore, the rules associated with these limit violations have hierarchical levels. Upon detection of a violation, KMES looks for persistence of the violation. If the mnemonic's value remains beyond limit boundaries, for a time greater than the persistence period, KMES

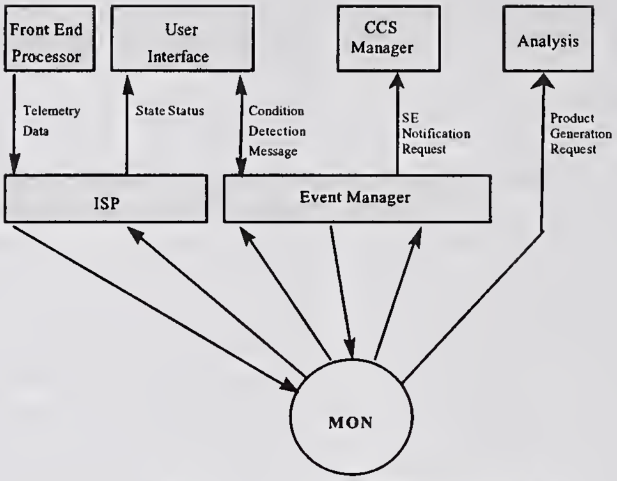


Figure-1 KMES External Interfaces

fires the associated contingency rules. As a part of actions within these rules, KMES sends requests to the Analysis subsystem for historical data products. Figure-1 shows KMES external Interfaces. KMES specifies the start time as well as the stop time for the requested data. Format of the requested data and type of the requested historical data are stored in ASCII files that are accessible to the Analysis subsystem. The generated historical data products are stored in a directory accessible to operators and system engineers. The system engineers (SEs) may use the data products to further analyze potential problems use the products.

4 KMES Architecture

Figure-2 shows process architecture for KMES. KMES consists of sub-processes for data communication as well as subprocesses for evaluation and reporting of the state of the spacecraft. The main processes are:

- RTD (Receive Telemetry Data)
- MGS (Manage States)
- REF (Respond to Events and Faults)
- PMD (Publish Monitoring Data)
- RMD (Route Monitoring Data), the RTserver

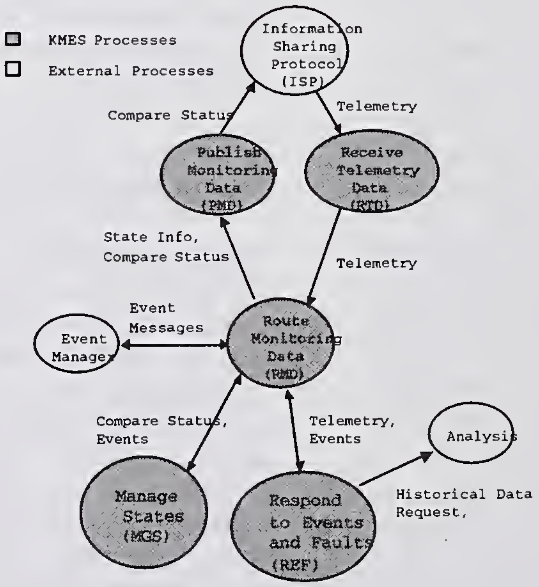


Figure-2 KMES Distributed Architecture

Originally, KMES employed a single database where all of KMES's processes included a copy of the data-base. However, this approach caused excess increase in the size of run time memory usage when KMES was expanded to include all seven subsystems of HST. Therefore, the design of KMES was modified to reduce the size of memory usage. In this new approach, every subprocess of KMES includes part of database that contains data related to mnemonics which are referenced or used within the subprocess. The following sections briefly describe function and features of each sub-process within KMES

4.1 Receive Telemetry Data (RTD)

The RTD process receives change-only data from the ISP server. RTD sends this data to the other KMES processes via the RMD process. Originally this subprocess employed RTdaq (a commercial product) and shared memory approaches for synchronization between RMD and ISP. However, it was found that occasionally data was dropped out during transmission between shared memory and RMD. In addition, RTdaq did not have capabilities to transmit status flags, which accompany the telemetry data. ISP sends the status flags as a part of data throughout the CCS subprocesses. These status flags indicate the status of data and they are set by the FEP subsystem within the CCS. A blank status flag indicates that the data is valid. At this time, ISP sends telemetry data with status flags that are set to nine possible values, one at a time. The status flag values are prioritized, four of these values indicate that the telemetry value is beyond pre-specified limits as defined in the PRD. The remainders of the status-flag values indicate if there has been a problem with data conversion or data transmission. The RTD subprocess was enhanced to eliminate the use of RTdaq as well as the shared memory approach. The enhanced version constructs custom designed data-packets that are in RTsmartSockets format. The packets contain changed only data and are sent to RTserver (RMD) for distribution to subprocesses within KMES.

4.2 Manage States (MGS)

The MGS process receives real-time telemetry data from ISP and generates compare status associated with each received Key Monitors mnemonic. The compare status indicates if there is a miscomparison (corresponding to a limit violation). This subprocess sets compare status in accordance with values of status flags that accompany telemetry data received from ISP. A compare status mnemonic may take four different values for a miscomparison corresponding to four possible ways of limit violations:

a) Yellow Low;
b) Yellow High;
c) Red Low;
d) Red High.

Yellow limits are warnings as specified by system engineers. Red limits are typically for serious violations associated with hardware limitations. MGS sends all compare status changes along with their status flags and time stamp to REF subprocess via RMD.

4.3 Respond to Events and Faults (REF)

REF includes all rules associated with limit violated Key Monitor mnemonics. Upon receipt of a miscomparison associated with Key Monitors from MGS, REF tracks the miscomparison for a pre-specified period of time (persistence time). If the limit violation persists, then REF fires the appropriate rules and sends appropriate historical data request with specified start time and stop time to the Analysis subsystem. REF also sends an event that indicates detection of the anomaly. The event message also indicates how soon the requested data product will be available for access by SEs or operators. The Analysis subsystem receives information for the data request from REF and retrieves the historical data in accordance with the format specified by SE(s) and Operators. The list and format of the data request are stored in specially designed files called "Historical Request Definition Files".

4.4 Publish Monitored Data (PMD)

The PMD process receives state data consisting of mnemonic's name, value, and time stamp from MGS (via RMD) while publishing this data to ISP. Along with this data there is a status flag indicating the override status of the mnemonic's value. The status flag indicates whether the user has overiden the mnemonic value within KMES or that the mnemonic value is derived by KMES.

4.5 Route Monitoring Data (RMD)

The RMD process consists of a real time RTserver. The process receives and routes data and event messages within KMES's processes.

5 KMES Characteristic Features

KMES consists of a group of distributed processes that communicate through a middleware layer. This modular design has many advantages such as maintainability and flexibility in where and how these processes are executed. If one process is overloading system resources, it can be relocated to another host machine. Among other advantages, KMES employs a distributed system approach to facilitate:

a) Change Data only executions
b) Maintenance simplification

This approach provides a capability to queue only those rules that are affected by the status of a mnemonic. In this way, only the rules that have to send a historical data request and e-mail or e-page will be fired and the rest of the rules will not be examined until later times when a status change affects them.

6 Results

KMES has been developed with rules associated with actions that are required when a Key Monitor has violated its limits. Currently, Rules associated with the following seven subsystems of HST have been implemented:

- Data Management Subsystem (DMS)
- Electrical Power Subsystem (EPS)
- Instrumentation & Communication Subsystem (I&C)
- Optical Telescope Assembly (OTA)
- Pointing Control Subsystem (PCS)
- Safing Subsystem (Safing)

The following section discusses the results obtained from operation of KMES during a simulated anomaly. For comparison, the results of the previously designed system, ESAM, for the same simulated anomaly is also demonstrated. As it was mentioned earlier, ESAM detects anomalies by comparing the engineering telemetry received from HST with some internally generated expected values. ESAM analyzes the discrepancies between the true and expected states to determine if an anomaly actually exists. Therefore, ESAM uses some tight and dynamically calculated limit boundaries. In contrast, KMES depends on some predefined and fixed limit boundaries.

The following section, compare the results from ESAM and KMES for a simulated scenario leading to failure of a sensor in the Electrical Power System of the spacecraft.

7 Scenario

The test scenario was designed to examine the rule executions resulted from an anomaly associated with one of the HST batteries. The spacecraft is equipped with six batteries. If only four batteries are nominal then the entire battery system is considered acceptable for normal operation. Previously captured data from a routine spacecraft orbit was fed into the HST simulator. The simulator was started in play back mode with continuous data feed. Figure-3 shows voltages for the first and the second batteries of the spacecraft, respectively. Figure-4 shows the currents associated with the first and second batteries. Figure-3 and Figure-4 also depict the high-expected limit and the low-expected limit calculated by ESAM. For comparison, Figure-3 also shows the constant limits used for Key Monitors out of bound violations. As shown, the constant limits are normally wider than the limits calculated by ESAM. The results for battery one demonstrates that the system was in normal operation until time 23:05, at this time (point A) a ramp down sensor anomaly was simulated into the telemetry data for voltage of the first battery. Figure-3 shows that within about 4 minutes and 40 seconds (point B) into the incident, the battery voltage fell below the low limit as calculated by ESAM. However, Figure-3 shows that after 9 minutes and 30 seconds into the anomaly, the battery voltage fell below the constant limit used by the FEP. At this time KMES received an associated status flag from FEP that indicated the limit violation. Therefore, KMES queued the rules associated with the battery anomaly and sent appropriate notifications and historical data requests when the limit violation persistence was satisfied. The results show that ESAM detected anomaly within about five minutes after initiation of the anomaly. However, KMES sent anomaly notifications within ten minutes after initiation of the anomaly.

8 Future Work

A well-structured distributed expert system to monitor the Key Monitors of the Hubble Space Telescope has been developed and delivered. The results for the first release of this system are presented. The following highlights some of the items sought for improvement and further enhancements of the monitoring system:

a) a variable persistence time for each or subset of the Key Monitor mnemonics;
b) retrieve appropriate operating procedure(s) for response associated with an anomaly;
c) track violation changes from yellow limit boundary into red limit violation boundary;
d) accept a user-defined periods for

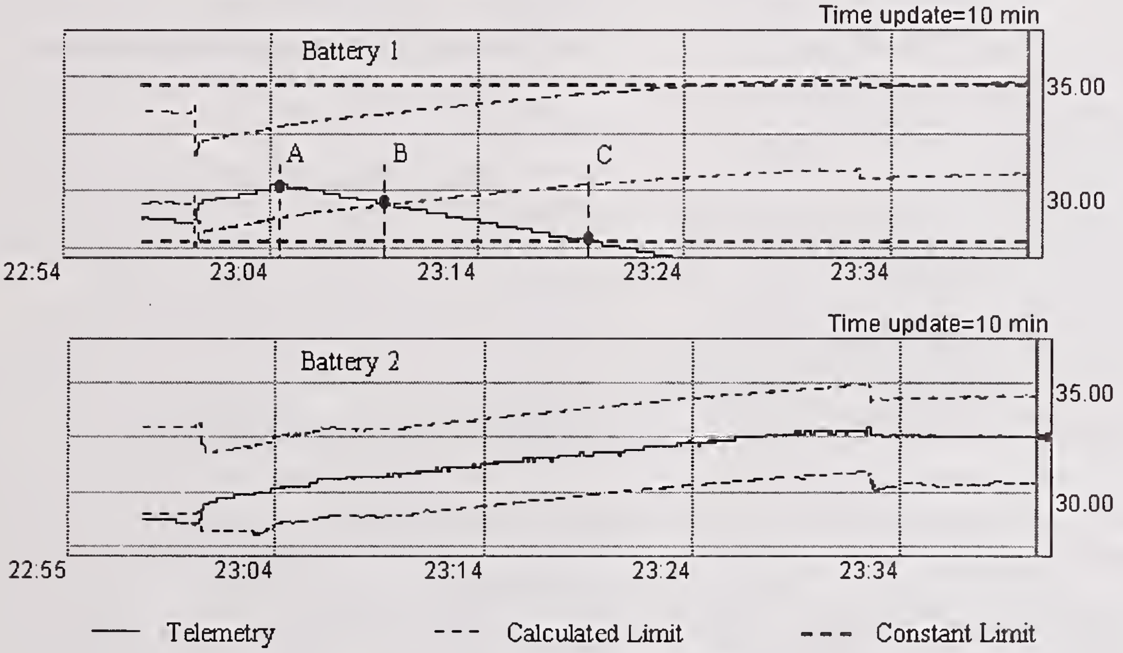


Figure-3 Results For Battery Voltage

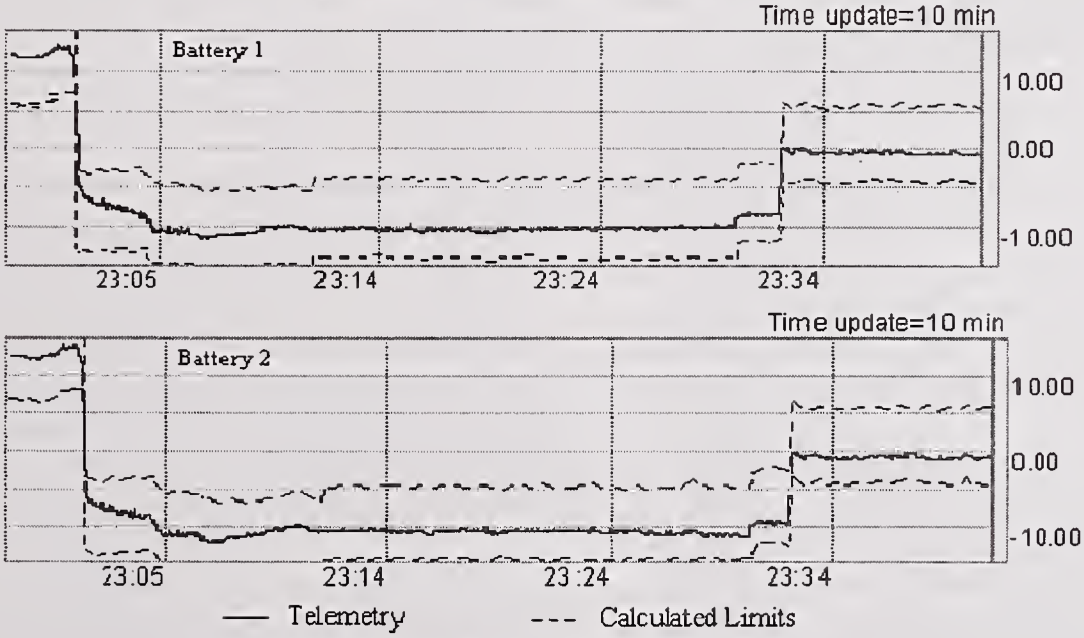


Figure-4 Results For Battery Current

notifications frequency associated with each of the limit violations;

e) GUI editor interface for visualization and graphical editing of rules.

9 Conclusion

A distributed expert system, KMES, for notification of anomaly and initial response (i.e., request associated engineering data for analysis) has been developed. The results of KMES for a simulated failure has been compared with similar results obtained from a previously designed expert system, ESAM. KMES uses the results of anomaly detection with constant limit values while ESAM calculates the expected limit boundaries. The results for detection of a sample sensor failure by the two systems are demonstrated. It was found that when calculated limits are employed then anomaly might be detected earlier than when constant limits are used for detection of the anomaly. However, notification of limit violations based on constant and established limits provides facilities for timely development of KB rules and execution of approved notifications.

10 Nomenclature

CCS	Command and Control System
ESAM	Expert System for Automated Monitoring
FEP	Front End Processor
ISP	Information Sharing Protocol
KMES	Key Monitors Expert System

11 References

[1] "A Real-Time Expert System for Automated Monitoring of the Hubble Space Telescope," R. Fakory and E. Ruberton, Intelligent Systems Conference, Gaithersburg, MD, September 1998.

[2] Consolidated HST Associated Mission Products (CHAMP) Contract, NAS50000.

[3] "Re-engineering of the Hubble Space Telescope (HST) Reduce Operational Costs (PartII)," M. Garvis, K. Lethonen and W. Burdick, internal report.

[4] "Knowledge Management Handbook," Jay Liebowitz, CRC Press 1999.

[5] "Development and Deployment of a Rule-Based Expert System for Autonomous Satellite Monitoring,"
L. Wong, F.Kronberg, A. Hopkins, F. Machi, P. Eastham, Astronomical Data Analysis, Vol 101, 1996.

[6] Talarian Corporation, support@talarian.com.

[7] "Hubble Space Telescope (HST) Contingency Plan document," Vol2, Lockheed Martin Missiles &
Space (LMMS), report No. LMSC/P061924, 1997.

Evaluating Performance for Distributed Intelligent Control Systems

Wayne J. Davis
General Engineering, University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA

Abstract

This paper provides a new framework for the distributed intelligent control of complex systems. The behavior of a given subsystem as it interacts with other subsystems is explored. The inherent limitations associated with distributed planning and control procedures are revealed. These limitations further limits one ability to evaluate system performance. Knowing these limitations, allows one to seek improved procedures for managing complex systems, which should also lead to improved system performance.

1. Introduction

Measuring system performance inherently represents a subjective task, beginning with the definition of the considered system. The decision of what system elements will be included within the considered system is arbitrary. Moreover, excluding elements from the considered system does not eliminate the potential for these elements to interact with the considered elements. Rather, such interactions become inputs to the considered system, whose values cannot be controlled. The definition of the considered system necessarily constrains the performance of the system because one must relinquish control of these environmental inputs.

Any performance criteria employed to evaluate a system must be based upon system variables that can be measured and controlled. Hence, the scope of the considered system inherently constrains the type of performance evaluations that can occur. Often there are multiple criteria to be considered, which necessitates compromise among the appropriate criteria. Compromise requires a subjective prioritization among the considered criteria, making absolute performance evaluations nearly impossible to achieve.

It becomes difficult to analyze and manage a complex system as a single monolithic entity. Complex systems are better represented as systems-of-systems. Again, the definition of the included subsystems is arbitrary. Furthermore, each included subsystem will have its own state and control variables. These variables again constrain which performance criteria can be considered by each subsystem. What often emerges is a collection of subsystems whose behaviors are characterized via different performance criteria.

Even if a monolithic specification for the system's planning and control problems can be made, there are still shortcomings, expecially since the monolithic approach ignores the system-of-systems nature. Monolithic specification do not capture the multi-resolutional nature of complex systems where given subsystems address system variables at different levels of detail and on different time scales. Monolithic approaches do not scale well. For large-scale systems, monolithic approaches become impossible to implement.

On the other hand, distributed planning and control introduces other problems. Today's distributed planning and control technologies do not capture the true nature of the distributed planning and control requirements for complex systems. Most decomposition procedures for distributing planning still assume that a monolithic planning problem exists (see Lasdon [1] and Wismer [2]). In general, this monolithic planning problem cannot be defined; and even if it could, its complexity would be well beyond the scope of problems that can be addressed with available decomposition procedures. Decomposition algorithms further seek an optimal solution to the monolithic planning problem. However, the relationship of optimal planning at the subsystem level toward the optimal planning for overall system within which it resides is simply not understood. Today, we do not know how to coordinate the planning at a subsystem in order to insure global optimality for the overall system within which the subsystem resides.

Control is essential to implement plans. Again, the current distributed control technologies are limited. Perhaps the most common distributed control procedure is the slow-fast decomposition (see Kokotovic et al. [3]). Slow-fast decompositions certainly can address situations where the desired response is known. They usually assume that an aggregated description for the overall response is known over an extended horizon, which includes the current time. Subsystems then manage the detailed description of this same trajectory over a shorter horizon which again includes the current time. This process continues where each subsystem addresses more detail over an even shorter horizon beginning with the current time. Implicitly, a monolithic control policy has been developed in that one assumes that the desired aggregate response is known over the entire time horizon.

Distributed intelligent control (distributed planning and control) approaches do not permit a monolithic description of the desired system trajectory. Rather, the system trajectory evolves as a collective response of several subsystems considering different temporal horizons and system elements. The planned response and the associated implementing actions evolve with time. Neither the monolithic planning or control problems are ever stated or solved. It is obviously difficult to manage such systems. Even more difficult is projecting their performance.

I revisited the distributed planning and control problem last year. My desire was to define what a distributed planning control system could accomplish. All the basic principles, including optimality and controllability, were set aside. The goal was to determine how a subsystem could address its assigned planning and control responsibilities while effectively interacting with other subsystems. Subsequently, the coordination of the interactions among the entire ensemble of distributed planning and control systems in order to provide an effective overall system response had to be addressed Testing effectiveness became a concern given the inherent inability to define the overall system problem as it continued to evolve in time.

This paper provides a brief discussion of the basic discoveries arising from this rapprochement. The fundamental principles of optimality and controllability have been reexamined and mathematical proofs/arguments do exist for the inherent limitations. Unfortunately, space limitations prevents me from providing these mathematical arguments. Instead, this paper will provide basic discoveries only. The paper first investigates how subsystems interact with each other. Next, the comprehensive nature of the overall system response arising from these interactions is addressed. Finally, the inherent limitations upon planning and control will be itemized. These limitations fundamentally impact one's ability to manage and project system performance. They must be addressed.

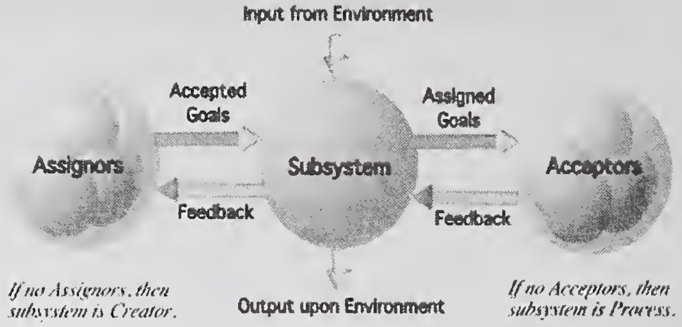


Figure 1: Basic interactions for a subsystem.

2. Fundamental Concepts

We begin our development with two basic assumptions:

- Most complex systems can be represented as a collection of subsystems that interact with each other. That is, complex systems are actually systems of subsystems.
- Each subsystem has a purpose, which it fulfills by executing tasks. Furthermore, the tasks that each subsystem can execute are related to the tasks that other subsystems can execute.

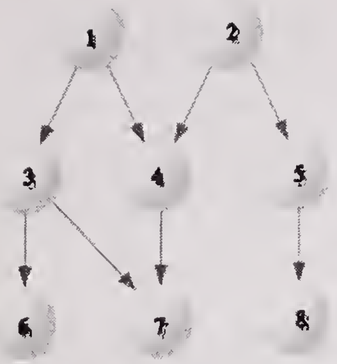
Consider a single subsystem. Its associated control inputs include (see Figure 1):

- Endogenous control inputs that it generates in order to implement its planned response.
- Assigned goals from other subsystems.
- Feedback information from other subsystems.
- Exogenous inputs from a subsystem's environment.

For a given subsystem, the assigned goals and feedback information might also be considered as exogenous inputs because these are generated by other subsystems. However, a given subsystem can determine which goals it will accept. The goals that it assigns to other subsystems will also influence their behavior and subsequent feedback information. Thus, only environmental inputs cannot be influenced in any manner by a given subsystem.

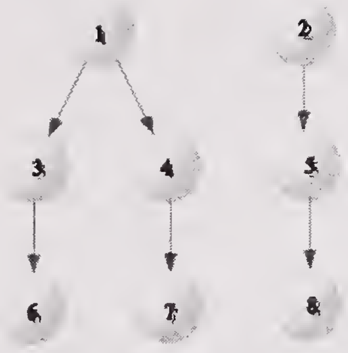
Should the subsystem have an option to accept or reject a goal? We believe that such an option is essential in order to insure that the recipient subsystem can feasibly respond to the goal. If one subsystem cannot satisfy its assigned goals, then the subsystem cannot respond in a feasible manner and the ability to control the subsystem is diminished or eliminated.

The assignment of goals to another subsystem represents one type of output that can occur as a subsystem responds to its control inputs. In addition, the given subsystem must provide feedback information to any other subsystem from which it has accepted a goal. Finally, the



Node	Assignors	Type
1	---	Creator
2	---	Creator
3	1	Coordinator
4	1, 2	Coordinator
5	2	Coordinator
6	3	Process
7	3, 4	Process
8	5	Process

Figure 2: Network representation of subsystem relationships.



Node	Assignors	Type
1	---	Creator
2	---	Creator
3	1	Coordinator
4	1	Coordinator
5	2	Coordinator
6	3	Process
7	4	Process
8	5	Process

Figure 3. Hierarchical system(s) where each subsystem (node) has at most one Assignor.

subsystem may also generate outputs that act upon the system's environment.

Typically, when one seeks to coordinate subsystems, one employs hierarchical based notions of supervisors (supremals) and subordinates (infimals). Hierarchies, right or wrongly, have been the subject of much recent criticism. In this paper, our desire is to provide a neutral atmosphere for discussing such coordination concerns.

We define the Assignors as the set of controllers that can assign goals to a given subsystem. Acceptors are the set of subsystems to which a subsystem can assign goals. Figure 1 depicts the proposed relationships among the subsystems.

There are two special situations. If the set of Assignors for a given subsystem is empty, then the subsystem receives only exogenous inputs from the its environment and feedback information from its Acceptors. We refer to such a subsystem as a Creator because it generates goals only, and does not accept any goals from any other subsystem. Every system model requires at least one creator. However, Creators are generally artificial constructs resulting from the modeling process. That is, the Assignors for a Creator are assumed to be outside the scope of the modeled system. Thus, goals coming from these external subsystems, or implicit Assignors, are viewed as inputs to a Creator from the system's environment.

If the Acceptors set for a given subsystem is empty, the subsystem is a Process. Processes can accept and process goals, but they cannot reassign their goals to any other subsystem. Hence, Processors can only accept inputs from their Assignors and the system's environment. In response to these inputs, they generate outputs upon the environment and provide feedback information to their Assignors.

We can graphically represent the proposed system structure (see Figure 2). We first define a node for each subsystem. We then employ directed arcs from a given subsystem's node to each node within its Acceptors set. Finally, from each subsystem node within a given subsystem's Assignors set, we draw a directed arc to the node for the given subsystem. Using network terminology, the Creator(s) become the source(s) to the system network while the Processes are the sinks.

In general, there can be more than one path from a given Creator to a given Process. (In Figure 2, there are multiple paths from node 1 to node 7.) However, there need not be a path from every Creator to every process. (In Figure 2, there is no path from node 1 to node 8.) In the special case where the number of elements in each subsystem's Assignors set is less than or equal to one, the representative system network becomes a tree and represents a conventional hierarchy (see Figure 3). If there is more than one Creator in the hierarchical case, then the overall system must be represented as a set of disjoint hierarchies that do not interact with each other (see Figure 3).

Although the potential for loops can exist within a system's network, loops should not exist from the conceptual point of view. Later we will show that the detail considered by an Acceptor is greater than its Assignor. We will also show that the planning horizon for any Acceptor should be less than that of the Assignor. Loops could occur when an Assignor for a given subsystem is also contained in the given subsystem's Acceptors set. However, if a subsystem is simultaneously contained within the Acceptors and Assignors sets for another given subsystem, then the simultaneous Acceptor/Assignor must be less detailed from the system to which it assigns goals and more detailed than the same system from which it accepts goals. Obviously, the two implied relationships are contradictory.

Therefore, we may conclude that the network representation of the relationships among subsystems for all meaningful systems' must be a directed acyclic network (containing no loops).

The reader should note that we have spoken of goal assignments rather than tasks, as mentioned earlier. We assume that a goal can contain a task. Moreover, a goal can also describe how an assigned task should be executed. For example, the Assignor might request that the task be completed by a given time or completed at minimum cost.

3. Toward an Integrated Approach

No (sub)system can generate an optimal response when acting as an independent agent. A given subsystem's response is dependent upon its goals and the subsequent response of the subsystems to which it has assigned goals. Furthermore, one cannot demonstrate that the collective response arising from the coordinated interaction among all its subsystems is optimal because we have not (and cannot) define the overall problem.

Because no subsystem can respond independently from the other subsystems, it follows that each subsystem must constantly interact with other subsystems: its Assignors and Acceptors. However, a given subsystem's interactions with an Assignor are fundamentally different from its interactions with an Acceptor. Each subsystem considers a time interval and a level of detail that differs from those of its Assignors and Acceptors. Each subsystem must move from its current state to a specified goal state while responding to any external inputs from the overall system's environment and any peculiarities that arise in its own dynamic evolution.

Let us consider the interaction of a given subsystem with its Assignors. A given subsystem can only address its behavior over an interval. Nevertheless, the way in which a subsystem responds within a time interval can affect the future behavior of the entire system beyond the considered time interval. The problem is that the given subsystem is incapable of assessing these future consequences beyond the time interval that it considers. The subsystem must instead rely upon the subsystems contained within its Assignors set to make such assessments. In performing this function, each Assignor considers the future in order to specify goals for the given subsystem. The subsystem receiving the goals employs those goals in order to define its desired final state at then end of its planning horizon.

Similarly, most subsystems are also limited by the level of detail that they can consider. In order to affect the more detailed responses that are required to meet its assigned goals, each subsystem assigns goals to its Acceptors. Thus, as each Acceptor addresses an assigned goal, it provides a more detailed system response on behalf of the subsystem that assigned the goal. The subsequent feedback information provided by the Acceptor during its execution of an assigned task assists the Assignor in assessing the beginning state for its planning horizon. (Later we will demonstrate that this beginning state for a subsystem's planning horizon *cannot* be the current system state. It must always be a projected future state from which the given subsystem will attempt to a desired final state.) Two extremes, or boundary conditions, for a given subsystem's planning/control problem have now been specified. Its included planning and control (intelligent control) capabilities then guide the given subsystem from its projected initial state toward the desired final state while responding to forecasted environmental inputs and other peculiarities of the system response.

Every element of the subsystem's planning/control problems changes with time. The subsystem's estimate of its initial state changes as its Acceptors execute their assigned tasks. The subsystem's goal changes with time as its Assignors respond to feedback information that the subsystem provides. Finally, the forecasts for the subsystem's future interactions with its environment must be constantly updated.

We can now define three basic functional requirements for each subsystem's intelligent controller. These include:

Task Accepting: The intelligent controller must interact with the intelligent controllers that manage each subsystem within its Assignors set. The purpose of this interaction is to define new goals and to update current goals. Each assigned goal specifies at least one task to be addressed along with a set of constraints. Because an Assignor addresses the system in a more aggregated sense than the subsystem that accepts the task, the Task Accepting function must decompose the assigned tasks into subtasks. In addition, the execution constraints accompanying each accepted task must also be reformulated in order to specify appropriate (or consistent) constraints for each defined subtask.

The task decomposition and the associated constraint specification comprise a goal decomposition process. This goal decomposition must guarantee that the accepted goals can be satisfied given the accepting subsystem's current state. The Task Accepting function is also responsible for continuously updating the projected response of the subsystem as feedback information to each Assignor. Remember, however, that an Assignor considers the system response in an aggregated manner. Thus, the Task Accepting function must summarize its projected response in order to provide the estimated performance statistics that can be understood by its Assignor.

Task Assigning: After the assigned goals are decomposed, the resulting subtasks and their associated execution constraints must be reassigned to the subsystem's Acceptors. In making the subsequent goal assignments, the Task Assigning function will employ the selected control law that implements the subsystem's current plan. The Task Assigning function also monitors feedback information from each Acceptor to which it has assigned a goal. Using this feedback information, it projects the future performance of the subsystem as it continues to execute its assigned goals under the selected control law. This projected response is then employed by the Task Accepting function within the same intelligent controller in order to provide feedback information to the subsystem's Assignors.

Performance Improvement: The system now has an estimated current state as well as a projected response as it implements its current goals under the planned response and enabling control law. The Performance Improvement Function continuously seeks a better control law for implementing the subsystem's assigned goals. Remember, however, that every element of the control problem is dynamic and uncertainties do exist. Given this reality, closed-loop control laws inherently perform best because they can tailor their response to the system's current state. It is also desirable to employ predictive control procedures whenever the current control action depends upon both the system's current and predicted future state. Whenever a new control law is selected, it is forwarded to Task Assigning function for implementation.

4. The Fundamental Principles of a Coordinated Response

This section addresses the basic system response. Figure 4 provides a primitive schematic for the multi-resolutional behavior of these systems. Let t_0 represent the current time that advances with real-time. We then divide the future time axis into several intervals, including $[t_1, t_2)$, $[t_2, t_3)$, $[t_3, t_4)$ and so forth. Note that we have not yet included a time interval between $[t_0, t_1)$ or $[t_4, \bullet)$ for reasons to be discussed later. In Figure 4, the entire state vector has been projected as a single value upon the y-axis. This state trajectory is further divided into segments: one segment for each time interval specified above. Let us assume that each segment corresponds to the trajectory for a given subsystem operating under the control of its intelligent controller. Considering the subsystem associated with the state trajectory on the interval $[t_3, t_4)$, its Assignors manage the state trajectory beyond t_4, while its Acceptors manage the state trajectory on the interval $[t_2, t_3)$.

Suppose we view each component of the state trajectory as a sophisticated "Slinky." The multi-resolutional nature of the systems implies that size and length of each "Slinky's" spring gets smaller and shorter as its associated time interval approaches t_0. Now let us further assume that two adjacent "Slinkies" are attached to each other and that the boundary conditions must match at each junction.

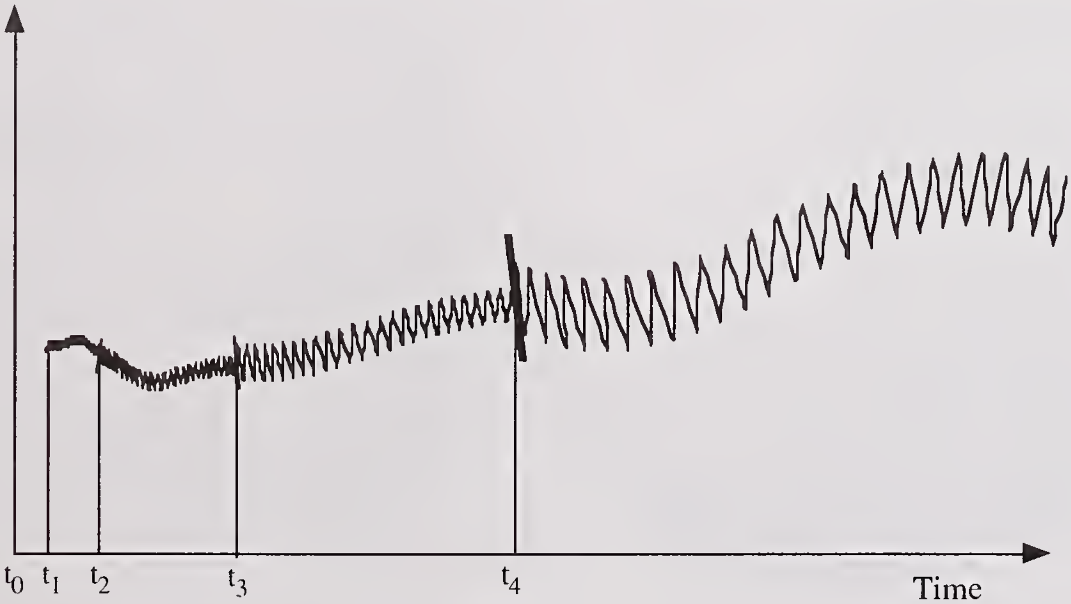


Figure 4: Basic Schema for multi-resolutional system response.

The "Slinky" for interval $[t_3, t_4)$ interfaces with a larger "Slinky," with less resolution, at t_4. It also interfaees with a shorter "Slinky," with greater resolution at t_3. Assume also that each "Slinky" can manage its shape. However, no "Slinky" can act independently of the others. Specifically, the "Slinky " on interval $[t_3, t_4)$ must interact with the "Slinky" operating beyond t_4 in order to define the boundary conditions at t_4. To be more preeise, the Task Accepting funetion of the $[t_3, t_4)$-subsystem must interact with the Tasking Assigning functions for the Assignors of the $[t_3, t_4)$-subsystem. Similarly, the Task Assigning function for the $[t_3, t_4)$-subsystem must interaet with the Task Aceepting functions of its Acceptors. Finally, the shape of the "Slinky" between t_3 and t_4 is controlled by the Performance Improvement function as the $[t_3, t_4)$-subsystem responds to forecasted external inputs that will likely act upon it during the interval $[t_3, t_4)$.

Given the recursive system-of-systems nature for the system, each Acceptor for the $[t_3, t_4)$-subsystem interacts with the $[t_3, t_4)$-subsystem in a manner similar to the way that the $[t_3, t_4)$-subsystem interacts with its Assignors. Similarly, each Acceptor for the $[t_3, t_4)$-subsystem will similarly interface with its own Acceptor's Task Accepting functions. Note also that none of the indicated subsystems ean touch t_0 because only a Process that has no Acceptors can reach t_0. (We will discuss this assertion later.)

Now, let us try to visualize the dynamic behavior of the proposed system's response. Remember, t_0 (the current time) must continue to advance in real time. We may assume that the entire state trajectory is dynamic, and neither the interface times ($t_1, t_2, \ldots$) nor the boundary conditions are fixed. Instead, the shared boundary conditions between two adjaeent "Slinkies" are constantly being renegotiated in real time. As the boundary conditions are modified and the forecasts for the external effects upon a given subsystem are updated, the intelligent controller responds by modifying the projected desired shape of the "Slinky" between the appropriate interface times.

We have stated that only Processes can affeet the system in real time. Several important conclusions follow:

- *No interface time at the junction of two subsystems' responses can ever occur.* These interface times constantly change with time and must always be greater than the current time t_0.
- *Only Processes react to real inputs from the external environment.* The other subsystems plan their response based upon forecasted inputs from the environment and their eurrent negotiated boundary eonditions.
- *The planned trajectories of the non-processing subsystems are never realized.* These planned trajeetories only eonjecture how the system will likely respond for planning purposes.
- *The purpose of the intelligent controllers for the non-processing subsystems is simply to establish goals for another subsystem.* The recursive system-of-system nature of these systems implies that these goals will become more detailed as their interfacing times approach t_0.

Figure 4 does not adequately depiet the interaction between a given subsystem and its Assignor(s) and Acceptors. In Figure 5, we provide a more detailed illustration of the proposed interaction among the subsystems as they interact with each other. It also illustrates the evolution of time and the limitations that a given system has in managing the response of the system.

Time advances from left to right in Figure 5. The large sphere represents the state space for the aggregate subsystem that projects into the most distant future. Within that subsystem's state space, there are two smaller spheres. The right-most of these spheres represents the goal space that the system seeks to reach at t_4. In this ease, we assume that the final goal is established by the system's environment because the Assignors for this subsystem have not been included within the system model. Note that this is an arbitrary choice based upon the modeler's desires and is determined to a eertain extent by how far the modeler wants to foreeast the system's future response.

The left-most sphere within the largest sphere represents the forecasted state at t_3 from which the aggregate subsystem initiates its planning. Thus, the aggregate subsystem represented by the largest sphere will plan on the interval $[t_3, t_4)$. The values for both t_3 and t_4 are dynamie and must increase with real-time, and t_4 is always greater than t_3. (The reader will note that we have not included t_4 within the subsystem's planning horizon because the goal state at t_4 is specified by an agent outside of the modeled system).

The role of the intelligent controller for the $[t_3, t_4)$-subsystem is to determine the ideal trajectory from the anticipated state at t_3 to the desired goal state at t_4. During that interval, the subsystem must also respond to other external inputs. Because the planning interval is beyond the current time, these external inputs must be foreeasted. Hence, the planned response on the $[t_3, t_4)$ interval is a projected response only. It will not (or cannot) be implemented as planned.

The $[t_3\text{-}t_4)$-subsystem cannot manage the response of the system before t_3 because it cannot address the detail required to describe the system's response prior to t_3. Rather, this detail will be addressed by two other subsystems as indicated by the second largest spheres in Figure 5. The fact that the spheres are smaller has no relation to the dimensions of each subsystem's state space. Rather, the diameters of the spheres correspond to the relative length of the planning interval that each subsystem addresses.

The $[t_3, t_4)$ subsystem estimates its initial state at t_3. Thus state is achieved by the subordinate's response on the $[t_2, t_3)$ time interval. In order to manage the subordinate subsystem's response, the $[t_3, t_4)$-subsystem must define goal states for the two subsystems at $t_3^{(1)}$ and $t_3^{(2)}$, respectively. However, the state variables considered by the subordinate subsystems are different than those considered by the $[t_3, t_4)$-subsystem. Hence, a transformation between the state spaces must occur. This transformation is implemented by the Task Assignor for the $[t_3, t_4)$-subsystem as it interacts with the Task Acceptors within $[t_3,t_4)$-subsystem's Acceptors. This transformation involves two types of interactions. With respect to the $[t_3, t_4)$-subsystem, the first interaction determines a mutually acceptable set of feasible goals for each Acceptor. The second interaction monitors each Acceptor's progress in achieving its assigned goals. Here, the Task Assigning function for $[t_3, t_4)$-subsystem must transform each Acceptor's projected goal achievement into the corresponding state representation within the $[t_3, t_4)$-subsystem's state space. Moreover, the individual Acceptor's response must be integrated to form a single composite estimate for the $[t_3, t_4)$-subsystem's initial state at t_3.

The goals established for the Acceptors will cover the time interval up to $t_3^{(1)}$ and $t_3^{(2)}$, respectively. In order to insure planning across the entire time interval up to t_4, it is essential that t_3 be less than or equal to either $t_3^{(1)}$ or $t_3^{(2)}$. Thus, the planning interval for a given Acceptor usually overlaps the planning interval of its Assignor(s). In addition, the state space for the individual Acceptors can also overlap each other. For example, it might be possible for both Acceptors to execute a given task. It is also possible that the state spaces are not entirely congruent. One Acceptor might be able to execute tasks that the other Acceptor cannot.

On the other hand, the state trajectories through the subsystem's state spaces must not intersect. Two distinct subsystems may not perform identical tasks upon the same entity at the same time. Two or more subsystems could possibly collaborate, but one subsystem would still assume primary control of the entity and subsystem actions upon the entity must differ from the others in some

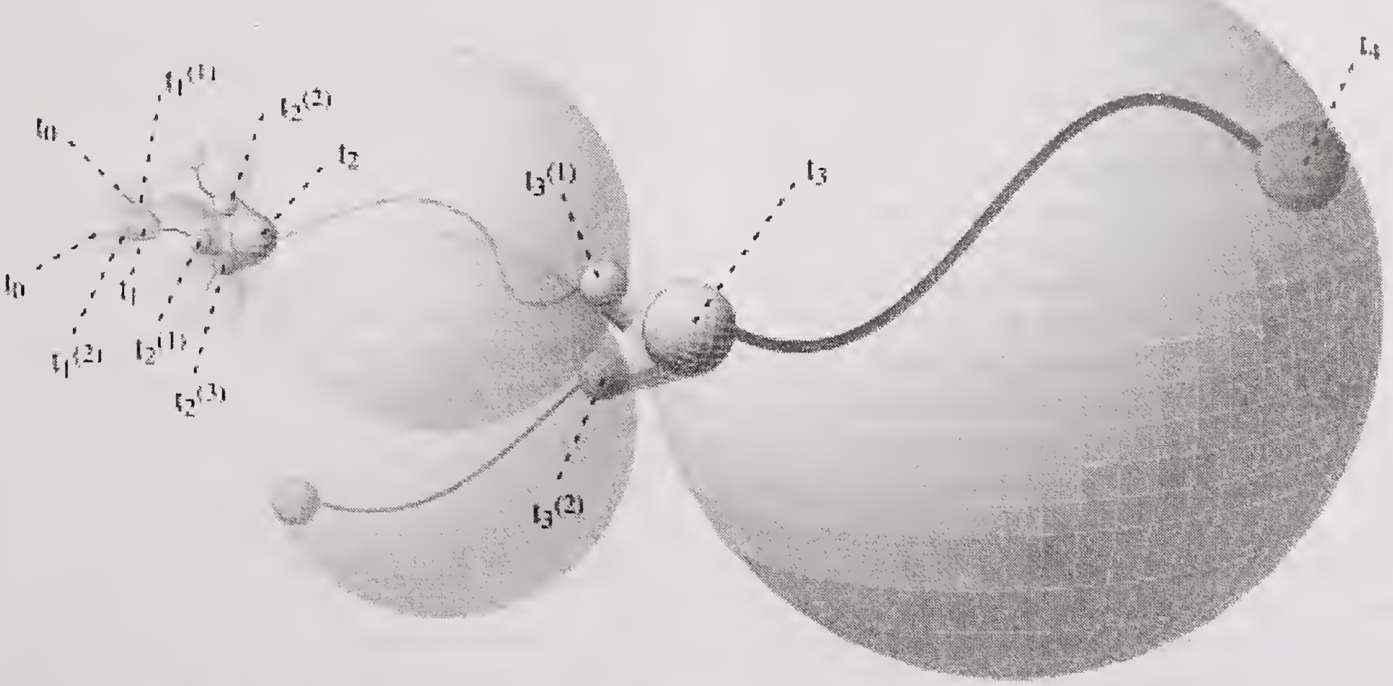


Figure 5. A more detailed representation of the multi-resolutional state evolution.

manner at a given time. A fundamental law of physics prevents two objects from occupying the same region of space and time simultaneously.

Given its desired final goal state at $t_3^{(1)}$, the $[t_2, t_3^{(1)})$-subsystem plans its response through its state space. The $[t_2, t_3^{(1)})$-subsystem interacts with its Acceptors in order to estimate its initial planning state at t_2. This interaction also establishes the goals for each of the $[t_2, t_3^{(1)})$-subsystem's Acceptors at $t_2^{(1)}$, $t_2^{(2)}$ and $t_2^{(3)}$. Having established each of their individual goals, the $[t_2, t_3^{(1)})$-subsystem's Acceptors can determine their individual initial planning states. The same process is repeated for each state trajectory emanating from t_4 until a process, (which has no Acceptors), is encountered. This terminating process can be managed at t_0. Hence, the recursive planning process generates a collection of state trajectories, each beginning at t_0 and terminating at t_4.

Figure 5 depicts a situation whcre hierarchical planning occurs. Each subsystem has at most one Assignor, and the collection of state trajectories illustrated in Figure 5 forms a tree. Observe that state trajectories continue to divide as the diagram progresses from the most distant time t_4 toward present time. Moreover, each subsystem has a single state trajectory to manage. In Figure 5, we did not include every possible subsystem in order to simplify the figure. (Observe that some of the state trajectories do not begin at t_0.) If all potential subsystems for a hierarchical system were includcd, then every path in the state-trajectory tree would start at t_0 and terminate at a common root at t_4.

Recently, hierarchical systems have fallen from favor. Certainly, hierarchies, like all organizational structures, do have their limitations. However, most limitations occur when the Assignors dictate their goal assignments and the Acceptors cannot reject an assigned goal. Recent management and distributed planning approaches seek to empower the subordinate subsystems with greater planning and control responsibilities. Contrary to popular belief, such empowerment does not negate hierarchical structures.

Unfortunately, there are situations where hierarchies are inappropriate. For example, the government typically seeks to prevent individual corporations from collaborating in order to create a monopolistic environment.

One benefit of the proposed approach is that we can now characterize the limitations that arise when one must employ a structure other than a hierarchy. In particular, we can now test many of the claims that the advocates of other architectures have cited.

5. Conclusions

The efficacy of current planning and control technologies requires a monolithic statement of the system's planning and control problems. If such monolithic statements cannot be made, then available planning and control technologies are probably irrelevant. The above discussion demonstrates that it will be impossible to provide such monolithic specifications for most complex systems.

One might question whether it is possible to provide a monolithic statement for any system's planning and control problems. Remember that one's definition of the system's boundary is arbitrary. In most cases, the defined system is still dependent upon other environmental subsystems that are being managed by other entities. A subsystem seldom has complete control over its planning and control responsibilities. Witout such control, it is impossible to demonstrate optimality of a planned /executed system response with respect to any performance criteria. Given the current state of affairs, performance evaluations for a given subsystem level or the composite system should be avoided. The primary goal must be to develop improved technologies for managing complex systems.

6. References

[1] L. S. Lasdon, Optimization Theory for Large systems, London, Macmillian Company, 1970.

[2] D. A. Wismer, Optimation Methods for Large-Scale Systems with Applications, New York, MCGraw-Hill Book Company, 1971.

[3] P. Kokotovic, H. Khalil and J. O'reilly, Singular Perturbation Methods in Control: Analysis and Design, New York, Academic Press, 1986.

Hypothesis Testing for Complex Agents

Joanna Bryson, Will Lowe† and Lynn Andrea Stein
MIT AI Lab
Cambridge, MA 02139
joanna@ai.mit.edu, las@ai.mit.edu

†Tufts University Center for Cognitive Studies
Medford, MA 02155
wlowe02@tufts.edu

Abstract

As agents approach animal-like complexity, evaluating them becomes as difficult as evaluating animals. This paper describes the application of techniques for characterizing animal behavior to the evaluation of complex agents. We describe the conditions that lead to the behavioral variability that requires experimental methods. We then review the state of the art in psychological experimental design and analysis, and show its application to complex agents. We also discuss a specific methodological concern of agent research: how the robots versus simulations debate interacts with statistical evaluation. Finally, we make a specific proposal for facilitating the use of scientific method. We propose the creation of a web site that functions as a repository for platforms suitable for statistical testing, for results determined on those platforms, and for the agents that have generated those results.

Keywords: *agent performance, complex systems, behavioral indeterminacy, replicability, experimental design, subjective metrics, benchmarks, simulations, reliability.*

1. Introduction

Humanoid intelligence is a complex skill, with many interacting components and concerns. Unless they are in an exceptional, highly constrained situation, intelligent agents can never be certain they are expressing the best possible behavior for the current circumstance. This is because the problem of choosing an ordering of actions is combinatorially explosive [9]. Consequently, for scientists or engineers evaluating the behavior of an agent, it is generally impossible to ascertain whether a behavior is optimal for that agent. Albus [2] defines intelligence as "the ability of a system to act appropriately in an uncertain environment, where appropriate action is that which increases the probability of success." Systems of such complexity are rarely amenable to proof-theoretic techniques [26]. In general, the only means to judge an increase in probability is to run statistical tests over an appropriately sized sample of the agent's behavior.

Computational systems, in contrast, are traditionally evaluated based on their *final* results and/or on their resource utilization [29]. The historical definition of computational process (c.f. Babbage, Turing, von Neumann) is modeled on mathematical calculation, and its validity is measured in terms of its ultimate product. If the output is correct — if the correct value is calculated — then the computation is deemed correct as well. More recent descriptions [e.g. 11] have added an assessment of the time, space, processor, and other resource utilization, so that a computation is only deemed correct if it calculates the appropriate value within some resource constraints.

This characterization of computation is less applicable when it comes to particular operating systems and other real-time computational systems. These systems have no final result, no end point summarizing their work. Instead, they must be evaluated in terms of ongoing behavior. Guarantees, where they exist, take the form of performance constraints and temporal invariants. Although formal analysis of correctness plays a role even in these systems, performance testing, including benchmarking, is an essential part of the evaluation criteria for this kind of computational system.

Computational agent design owes much to computer science. But the computationalist's tendency to evaluate in terms of ultimate product is as inappropriate for computational agents as it is for operating systems. Instead, metrics must be devised in terms of ongoing behavior, performance rather than finitary result. But what is the analog to benchmarking when the tasks are under-specified, ill-defined, and subject to interpretation and observer judgment?

In this paper, we will examine issues of running such evaluations for complex agents. By *complex agents* we mean autonomous agents such as robots or VR characters capable of emulating humanoid or at least vertebrate intelligence. We will discuss hypothesis testing, including the statistical controversies that have lead to the recent revisions in the standard experimental analysis endorsed by the American Psychological Association. We will also discuss recent advances in methodologies for establishing quantitative metrics for matters of human judgment, such as whether one sentence is more or less grammatical than another, or an anecdote is more or less appropriate. We propose a means to facilitate hypothesis testing between groups: a simulation server running a number of benchmark tests.

2. Motivation: Sources of Uncertainty

Although there is certainly a role for using formal methods in comparing agent architectures [e.g. 8, 6], what we as agent designers are ultimately interested in is comparing the resulting *behavior* of our agents. Given the numerous complex sources of indeterminacy in this behavior, such comparison requires the application of the same kind of experimental methodology that has been developed by psychology to address similar problems. In this section we review some of the sources of this indeterminacy; in the next we will review analytic approaches for addressing them.

The first source of indeterminacy is described above: The combinatorial complexity of most decision problems makes absolute optimality an impractical target. Thus even if there is a single unique optimal sequence of actions, in most situations we cannot expect an agent to find it. Consequently, we will expect a range of agents to have a range of suboptimal behaviors, and must find a way of comparing these.

The next source of indeterminacy is the environment. Many agents must attempt to maintain or achieve multiple, possibly even contradictory goals. These goals are often themselves uncertain. For example, the difficulty of eating is dependent on the supply of food, which may in turn be dependent on situations unknowable to the agent, whether these be weather patterns, the presence or absence of other competing agents, or in human societies, local holidays disrupting normal shopping. Thus in evaluating the general efficacy of an agent's behavior, we would need a large number of samples across a range of environmental circumstances.

Another possible source of indeterminacy is the development of agents. As engineers, we are not really interested in evaluating a single agent, but rather in improving the state-of-the-art in agent design. In this case, we are really interested in what approaches are most likely to produce successful agents. This involves uncertainty across development efforts, complicated by individual differences between developers. Many results contending the superiority or optimality of a particular theory of intelligence may simply reflect effective design by the practitioners of that theory [e.g. 7].

Finally, the emphasis of this workshop is on natural, human-like behavior. Humans are highly social animals, and social acceptability is an important criteria for intelligent agents. However, sociability is not a binary attribute: it varies in degrees. Further, a single form of behavior may be considered more or less social by the criteria of various societies. Evaluations of systems by such criteria requires measurement over a population of judges.

3. Current Approaches to Hypothesis Testing

The previous section presented a number of challenges to the evaluation of complex, humanoid agent building techniques. In this section we review methodologies used by psychology — the evaluation of human agents — that are available to address these challenges.

Although it is obvious that comparing two systems requires testing, the less obvious issues are how many tests need to be run and what statistical analysis needs to be used in order to answer these questions. In this section we describe three increasingly common problems in Artificial Intelligence and discuss a set of experimental techniques from the behavioral sciences that can be used to address them.

The first problem is variability in results: We need to know whether performance differences that arise over test replications can be ascribed to varying levels of a system's ability or to variation in lighting conditions, choice of training data, starting position, or some other or some other external (and therefore uninteresting) source. Psychology uses statistical techniques such as the Analysis of Variance (ANOVA) to address these issues. The second problem is of disentangling complex and unexpected interactions between subparts of a complex system. This can also be addressed using ANOVA coupled with factorial experimental design. The third problem is that of rigorously and meaningfully evaluating inherently subjective data. Since many psychology experiments investigate inherently subjective matters, the field has developed a set of techniques that will be of use to artificial agent designers as well. The next three sections describe these solutions in more detail.

3.1 Variability in Results

The problem of comparing performance variability due to differences in ability and variability due to extraneous factors is ubiquitous in psychology. It is dealt with by procedures known collectively as Analysis of Variance or ANOVA.

3.1.1 Standard ANOVA

In a typical experimental design for comparing performance, K systems are tested N times each. If the variation in performance between the K systems outweighs the variability among each system's N runs, then the system performances are said to be *significantly different*. We then examine the systems pairwise to get information about ordering. The ANOVA allows us to infer that e.g. although there are differences overall between the K=4 systems (i.e. some are better than others), the performance difference between 3 and 4 is reliable, whereas the difference between 1 and 2 is not reliable because it is outweighed by the amount of extraneous variation across the N tests. In this case, although 1 may perform on average better than 2, this does not imply that it is actually better on the task. If the experiment were repeated then 2 would have reasonable chance of performing on average the same as 1, or even better.

The notion of *reasonable chance* used above is the essence of the concept of significant difference. System 3 is on average better than 4 in this experiment and the ANOVA tells us that performances are significantly different at the .05 level (expressed as p<.05). This means that in an infinite series of experimental replications, if 3 is in fact exactly *as good* as 4, i.e. there is no genuine performance difference, then the probability of getting a performance difference as large or larger than the one observed in this experiment is 0.05. The smaller this probability becomes, the more reliable the difference is. In contrast, the fact that the average performances of 1 and 2 are not significantly different means their ordering in this experiment is not reliable because there is a more than 0.05 probability that the ordering would not be preserved in a replication.

Notice that hypothesis testing using ANOVA does not *guarantee* an ordering, it presents probabilities that each part of the ordering is reliable. This is a fundamental difference between experimental evidence and proof. Scientific method increases the probability that hypotheses are correct but it does not demonstrate them with complete certainty.

The binary output of hypothesis tests (significant difference versus no significant difference) and its probability is an unnecessarily large loss of information. The American Psychological Association have consequently recently moved to emphasize confidence intervals over simple hypothesis testing. A *confidence interval* is a range, centered on the observed difference, that in the hypothetical replications will contain the true performance value some large percentage, say 95%, of the time. In the example above, each system has a 95% confidence interval, or *error bar*, centered on its average performance with width determined by the amount of variability between runs. When two intervals overlap, there is a significant probability that a replication will not preserve the current ordering among the averages and we can conclude that the corresponding performance difference is unreliable. This method gives the same result as simple hypothesis testing above — the performances are not significantly different — but is much more informative: confidence intervals give an idea about how much variability there is in the data itself and yield a useful graphical representation of analytical results.

3.1.2 Alternative Approaches to Analysis

Stating confidence intervals is more informative than simple significance judgments. However, it also relies on an hypothetical infinity of replications of an experiment. This aspect of classical statistical inference is a result of assuming that the true difference in performance is fixed and the observed data is a random quantity. Alternatively, in Bayesian analysis the difference is considered uncertain and is modeled as a random variable whereas the results are fixed because they have already been observed [5]. The result is a probability distribution over values of the true difference. To summarize the distribution an interval containing 95% of the probability mass can be quoted. This takes the same form as a confidence interval, except that its interpretation is much simpler: Given the observed results, the probability that the true difference is in the interval is 0.95, so if the interval contains 0, there is a high probability that there is no real performance difference between systems.

The Bayesian approach makes no use of hypothetical experimental replications and is more naturally extended to deal with complicated experimental designs. On the other hand, it does require an initial estimate (or prior distribution) for the probabilities of various values of the performance difference before seeing test data. There is much controversy about which of these approaches is more appropriate. In the context of AI however, we need not take a stand on this issue. The two approaches answer different questions, and for our purposes the questions answered by classical statistics are of considerable interest. Unlike many of the natural sciences, the performance of AI systems over multiple replications is not only accessible, but of particular interest. To the extent we are engineers, AI researchers must be interested in reliability and replicability of results.

3.2 Testing for Interacting Components

Many unpleasant software surprises arise from unexpected interactions between components. Unfortunately, in a complex system it is typically infeasible to discover the nature of interactions analytically in advance. Consequently *factorial experimental design* is an important empirical tool.

As an example, assume that we can make two changes A and B to a system. We could compare the performance of the system with A to the same system without it, using the ANOVA methods above, and then do the same for B. But when building a complex system it is essential to also know how A and B affect performance together. Separate testing will never reveal, for example, that adding A generates a performance improvement only when B is present and not otherwise. This is referred to as an *interaction* between A and B, and can be dealt with by testing all combinations of system additions, leading to a factorial experiment. Factorial experiments are analyzed using simple extensions to ANOVA that test for significant interactions as well as simple performance differences. Factorial ANOVA methods are described in any introductory statistics textbook [e.g 23].

In the discussion above we have implicitly assumed that differences in performance can be modeled as continuous quantities, such as distance traveled, length of conversation or number of correct answers. When the final performance measure is discrete, e.g. success or failure, then *logistic regression* [1, ch.4] is a useful way to examine the effects of additions or manipulations on the system's success rate. Information about the effects of arbitrary numbers of additions, both individually and in interaction, is available using this method, just as in the factorial ANOVA. Logistic regression also gives a quantitative estimate of *how much* the probability of success changes with various additions to the system, which gives an idea of the importance of each change.

3.3 Quantifying Inherently Subjective Data

Often performance evaluation involves judgments or ratings from human subjects. Clearly it is not enough that one subject judges an AI conversation to be lifelike because we do not know how typical that subject is, and how robust their opinion is. It would be better to choose a larger sample of raters, and to check that their judgments are reliable. When ratings are discrete (good, bad) or ordinal (terrible, bad, ok, good, excellent) then Kappa [22] is a measure of between-rater agreement that varies from 1 (perfect agreement) to -1 (chance levels of agreement). For judgments of continuous quantities the intraclass correlation coefficient [13] performs the same task.

However, such discrete classifications are often clumsy. Because a rating system is itself subjective, the extra variance added by difference in interpretation of a category can lose correlations between subjects that actually agree on the relative validity or likeability of two systems. Further, we would really prefer in many circumstances to have a continuous range of difference values. Such results can be provided by *magnitude estimation*, a technique from psychophysics. For example, Bard *et al.* [4] have recently introduced the use of magnitude estimation to allow subjects to judge the acceptability of sentences which have varying degrees of syntactic propriety. In a magnitude estimation task, each subject is asked to assign an arbitrary number as a value for the first example they see. For each subsequent example, the subject need only say how much more or less acceptable it is, with reference to the previous value, e.g. twice as acceptable, half as acceptable and so on. This allows subjects to pick a scale they feel comfortable with manipulating, yet gives the experimenter a generally useful metric. For example, in Bard *et al.*'s work, a subject might give the first sentence an 8, the next a 4, the following a 32 — the experimenter records 1s, .5s and 4s respectively. This method has been shown to reduce the number of judgments necessary to get very reliable and accurate estimates of acceptability, relative to other methods.

Bard *et al.* manipulate the sentences themselves, but it is clear that magnitude estimation can equally well be used to get fine-grained judgments about how natural the output of a natural language processing (NLP) system is, and the degree to which this is improved by adding new components. Nor is the method limited to linguistic judgments, for it should be equally effective for evaluating ease of use for teaching software, the psychological realism of virtual agents or the comprehensibility of output for theorem proving machinery.

4. Environments for Hypothesis Testing: Robots and Simulations

As the previous sections indicate, one of the main attributes of statistically valid comparisons is a large number of experimental trials. Further, these experimental conditions should be easily replicable and extendible by other laboratories. In Section 5. we propose that a good way to facilitate such research is to create a web location dedicated to providing source code and statistics for comparative evaluations over a number of different benchmark tasks. This has approach has proven useful in neural network research, and should also be useful for complex agents. However, it flies in the face of one of the best-known hypotheses of complex agent research: that good experimental method requires the use of robots. Consequently, we will first provide an updated examination of this claim.

4.1 Arguments Against Simulation

Simulation is an attractive research environment because it is easy to maintain valid controls, and to execute large numbers of replications across a number of machines. However, there have been a number of important criticisms leveled against this approach.

A Simulations never replicate the full complexity of the real world. In choosing how to build a simulation, the researcher first determines the 'real' nature of the problem to be solved. Of course, the precise nature of a problem largely determines its solution. Consequently, simulations are not valid for truly complex agents, because they do not test the complete range of problems a natural or embodied agent would face.

B If a simulation truly were to be as complicated as the real world, then building it would cost more time and effort than can be managed. It is cheaper and more efficient to build a robot, and allow it to interact with the real world. This argument assumes one of basic hypotheses of the behavior-based approach to AI [3], that intelligence is by its nature simple and its apparent complexity only reflects the complexity of the world it reacts to. Consequently, spending resources constructing the more complicated side of the system is both irrational and unlikely to be successful.

C When researchers build their own simulations, they may deceive either themselves or others as to the validity or complexity of the agents that operate in it. Since both the problem and the solution are under control of the researcher, it is difficult to be certain that neither unconscious nor deliberate bias has entered into the experiments. In contrast, a robot is considered to be clear demonstrations of autonomous artifact; its achievements cannot be doubted, because it inhabits the same problem space we do.

4.2 *Are Robots Better than Simulations?*

These arguments have led to the wide-spread adoption of the autonomous robot as a research platform, despite the known problems with the platform [16]. These problems reduce essentially to the fact that robots are extremely costly. Although their popularity has funded enough research and mass production to reduce the initial cost of purchase or construction, they are still relatively expensive in terms of researcher or technician time for programming, maintenance, and experimental procedures. This has not prevented some researchers from conducting rigorous experimental work on robot platforms [see e.g. 10, 25]. However, the difficulty of such procedures adds urgency to the question of the validity of experiments in simulation.

This difficulty has been reduced somewhat by the advent of smaller, more robust, and cheaper mass-produced robot platforms. However, these platforms still fall prey to a second problem: mobile robots do not necessarily address the criticisms leveled above against simulations better than simulations do. There are two reasons for this: the need for simplicity and reliability in robots, and the growing sophistication of simulations.

The constraints of finance, technological expertise and researchers' time combine to make it extremely unlikely that a robot will operate either with perception anything near as rich as that of a real animal, nor with actuation having anything like the flexibility or precision of even the simplest animals. Meanwhile, the problem of designing simulations with predictive value for robot performance has been recognized and addressed as a research issue [e.g. 18]. All major research robot manufacturers now distribute simulators with their hardware. In the case of Khepera, the robot most used by researchers running experiments requiring large numbers of trials, the pressure to provide an acceptable simulator seems to have not only resulted in an improved simulator, but also a simplified robot, thus making results on the two platforms nearly identical. Clearly this similarity of results either validates the use of the Khepera simulator, or invalidates the use of the robot.

When a simulator is produced independent of any particular theory of AI as a general test platform, it defeats much of the objection raised in charges *A* and *C* above, that a simulator is biased towards a particular problem, or providing a particular set of results. In fact, complaint *C* is particularly invalid as a reason to prefer robotics. Experimental results provided on simulations can be replicated precisely in other laboratories. Consequently, they are generally *more easily* tested and confirmed than those collected on robots. To the extent that a simulation is created for and possibly by a community — as a single effort resulting in a platform for unlimited numbers of experiments by laboratories world-wide, that simulation also has some hope of overcoming argument *B*.

This gross increase in the complexity of simulations has particularly true of two platforms. First, the simulator developed for the simulation league in the RoboCup soccer competition has proven enormously successful. Although competition also takes place on robots, to date the simulator league provides far more "realistic" soccer games in terms of allowing the demonstration of teamwork between the players and flexible offensive and defensive strategies [21, 19]. This success has encouraged the RoboCup organization to tackle an even more complex simulator designed to replicate catastrophic disasters in urban settings [20]. This simulator is intended to be sufficiently realistic as to eventually allow for swapping in real-time sensory data from disaster situations, in order to allow disaster relief to monitor and coordinate both human and robotic rescue efforts.

The second platform is also independently motivated to provide the full complexity of the real world. This is the commercial arena of virtual reality (VR), which provides a simulated environment with very practical and demanding constraints which cannot easily be overlooked. Users of virtual reality bring expectations from ordinary life to the system, and any agent in the system is harshly criticized when it fails to provide adequately realistic behavior. Thórisson [30] demonstrates that users evaluate a humanoid avatar with which they have held a conversation as much more intelligent if it provides back-channel feedback, such as eyebrow flashes and hand gestures, than when it simply generates and interprets language. Similarly Sengers [27] reviews evidence that users cannot become engaged by VR creatures operating with overly reactive architectures, because the agents do not spend sufficient time telegraphing their intentions or deliberations. Such constraints have often been overlooked in robotics.

In contrast, robots which must be supported in a single lab with limited technical resources are likely to deal with far simpler tasks. Robots may face far fewer conflicting goals, lower time-related conflicts or expectations, and even fewer options for actuation. Although robots still tend to have more natural perceptual problems than simulated or VR agents, even these are now increasingly being addressed with reliable but unnatural sensors such as laser range finders.

4.3 *Roles for Robots and Simulations*

Robots are still a highly desirable research platform. They provide complete systems, requiring the integration of many forms of intelligence. Many of the problems they need to solve are closely related to animal's problems, such as perception and navigation. In virtual reality, perfect perception is normally provided, but motion often has added complication over that in the real world. Depending on the quality of the individual virtual reality platform, an agent may have to deliberately not pass through other objects or to intentionally behave as if it were affected by gravity or air resistance. Even in the constantly improving RoboCup soccer simulator, there are outstanding difficulties in simulating important parts of the game, such as the goalkeeper's ability to kick over opposing team members (currently compensated for by allowing the keeper to "warp" to any point in the goal box instantaneously when already holding the ball.)

Robots being embodied in the real world are still probably the best way to enforce certain forms of honesty on a researcher. A mistake cannot be recovered from if it damages the robot, an action once executed cannot be revoked. Though this is also true of some simulations [e.g. 31], particularly in the case of younger students, these constraints are better brought home on a robot, as it becomes more apparent why one can't 'cheat.' Finally, building intelligent robots is a valid end in itself. Commercial intelligent robots are beginning to prove very useful in care-taking and entertainment, and may soon prove useful in areas such as construction and agriculture. In the meantime robots are highly useful in the laboratory for stirring interest and enthusiasm in students, the press and funding agencies. However, given the arguments above, we conclude that the use of robots as experimental platforms is neither necessary nor sufficient in providing evidence about complex agent intelligence. Robots, like simulations, must be used in combination with rigorous experimental technique, and even so can only provide evidence, not conclusive proof, of agent hypotheses.

In summary, neither robots nor simulation can provide a single, ultimate research platform. But then, neither can any other single research platform or strategy [15]. While not denying that intelligence is often highly situated and specialized [14, 17], to make a general claim about agent methodology requires a wide diversity of tasks. Preference in platforms should be given to those on which multiple competing hypotheses can be tested and evaluated, whether by qualitative judgments such as the preference of a large number of users, or by discrete quantifiable goals to be met, such as a genetic fitness function, or the score of a soccer game.

5. Coordinating Hypothesis Testing

Whether there can be general solutions to problems of intelligence is an empirical matter that has already been tested in some domains. For neural networks and other machine learning methods, the UCI Machine Learning Repository holds a large collection of benchmark learning tasks. Besting these benchmarks is not a necessary requirement for the publication of a new algorithm, but showing a respectable performance on them improves the reception of new contributions. Essentially, benchmarks are one indication for both researchers and reviewers of when an innovation is likely to be of interest.

Further, Neal and colleagues at the University of Toronto have constructed DELVE [24], a unified software framework for benchmarking machine learning methods. DELVE contains a large number of benchmark data sets, details of various machine learning techniques, currently mostly neural networks and Gaussian Processes, and statistical summaries of their performance on each task. One of the most important requirements is that each method is described in enough detail that it could be implemented by another researcher and would obtain a similar performance on the tasks. This ensures that the mundane but essential decisions that are an essential part of many learning algorithms (e.g. setting weight decay parameters, choosing k in k-nearest-neighbor rules) are not lost.

We propose a complex agent comparison server or web site, to be at least partially modeled on DELVE. This site should allow for the rating of both agent approaches and comparison environments, thus encouraging and facilitating research in both fields. It could also be annotated for educational purposes, indicating challenges and environments well suited to school, undergraduate, and graduate course projects. Such a site might provide multiple indices, such as:

- Environments, ranked by number and/or diversity of participants.
- Agent architectures (e.g. Soar, Behavior-Based AI). This should also allow for the petition for new categories.
- Contestants and/or contesting labs or research groups . This allows researchers interested in a particular approach to see any related work. Ranked by the number and/or diversity of environments.

Here are some examples of already existent platforms which might be included on the server:

- RoboCup [21, 19].
- Khepera robot competitions. Both of these two suggestions provide simulations as well as organized robotic competitions. They test learning and perception as well as planning or action selection.
- Tile World and Truck World, designed as complex planning domains. [15]
- Tyrrell's Simulated Environment [31] designed to test action-selection and goal management.
- Chess.
- An analog Turing Test, using magnitude estimation to compare dialog systems.

In addition, there are at least two software environments designed specifically to allow testing and comparison of a number of different architectures, though they contain no specific experimental situations as currently developed. These environments are Cogent [12] and the Sim_agent Toolkit [28].

6. Conclusion

To summarize, we believe that as agents approach the goal of being psychologically realistic and relevant, their evaluation will require the techniques that have been developed in the psychological sciences. This evaluation is critical in providing a gradient as we search for the right sorts of techniques to build complex agents. The techniques of hypothesis testing have been refined to describe truly complex agents. However, these are scientific techniques, not proofs. They do not give us certain answers, only more information. We believe many of the criticisms of benchmark testing made in the past failed to properly acknowledge this feature of experimentation. We should trust increased probability, rather than proof-theoretic guarantees. The more people perform tests across competing hypotheses, the more likely we will be to achieve our research goals, whether they are engineering complex, social agents, or understanding the nature of intelligence.

Acknowledgments

The authors would like to acknowledge early discussions with Brendan McGonigle and Ulrich Nehmzow on this topic.

References

[1] A. Agresti. *Categorical Data Analysis*. John Wiley and Sons, 1990.

[2] J. S. Albus. Outline for a theory of intelligence. *IEEE Transactions on Systems, Man and Cybernetics*, 21(3):473–509, 1991.

[3] Ronald C. Arkin. *Behavior-Based Robotics*. MIT Press, Cambridge, MA, 1998.

[4] E. Bard, D. Robertson, and A. Sorace. Magnitude estimation of linguistic acceptability. *Language*, 72(1):32–68, 1996.

[5] G. E. P. Box and G. C. Tiao. *Bayesian inference in statistical analysis*. Addison-Wesley, Reading, Massachusetts, 1993.

[6] Joanna Bryson. Cross-paradigm analysis of autonomous agent architecture. *Journal of Experimental and Theoretical Artificial Intelligence*, 12(2):165–190, 2000.

[7] Joanna Bryson. Hierarchy and sequence vs. full parallelism in reactive action selection architectures. In *From Animals to Animats 6 (SAB00)*. MIT Press, 2000.

[8] Joanna Bryson and Lynn Andrea Stein. Architectures and idioms: Making progress in agent design. In *The Seventh International Workshop on Agent Theories, Architectures, and Languages (ATAL2000)*, 2000. to be presented July 2000.

[9] David Chapman. Planning for conjunctive goals. *Artificial Intelligence*, 32:333–378, 1987.

[10] David Cliff, Philip Husbands, and Inman Harvey. Explorations in evolutionary robotics. *Adaptive Behavior*, 2(1):71–108, 1993.

[11] S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on the THeory of Computing*, pages 151–158, New York, 1971. Association for Computing Machinery.

[12] R. Cooper, P. Yule, J. Fox, and D. Sutton. COGENT: An environment for the development of cognitive models. In U. Schmid, J. F. Krems, and F. Wysotzki, editors, *A Cognitive Science Approach to Reasoning, Learning and Discovery*, pages 55–82. Pabst Science Publishers, Lengerich, Germany, 1998. see also http://cogent.psyc.bbk.ac.uk/.

[13] P. E. Fleiss and J. L. Shrout. Intraclass correlations: Uses in assessing rater reliability. *Psychological Bulletin*, 86(2):420–428, 1979.

[14] C.R. Gallistel, Ann L. Brown, Susan Carey, Rochel Gelman, and Frank C. Keil. Lessons from animal learning for the study of cognitive development. In Susan Carey and Rochel Gelman, editors, *The Epigenesis of Mind*, pages 3–36. Lawrence Erlbaum, Hillsdale, NJ, 1991.

[15] Steve Hanks, Martha E. Pollack, and Paul R. Cohen. Benchmarks, testbeds, controlled experimentation and the design of agent architectures. Technical Report 93–06–05, Department of Computer Science and Engineering, University of Washington, 1993.

[16] Ian D. Horswill. *Specialization of Perceptual Processes*. PhD thesis, MIT, Department of EECS, Cambridge, MA, May 1993.

[17] Ian D. Horswill. *Specialization of Perceptual Processes*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, May 1993.

[18] N. Jakobi. Evolutionary robotics and the radical envelope of noise hypothesis. *Journal Of Adaptive Behaviour*, 6(2):325–368, 1997.

[19] Hiroaki Kitano. Special issue: Robocup. *Applied Artificial Intelligence*, 12(2–3), 1998.

[20] Hiroaki Kitano. Robocup rescue: A grand challenge for multiagent systems. In *The Fourth International Conference on MultiAgent Systems (ICMAS00)*, pages 5–12, Boston, 2000. IEEE Computer Society.

[21] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, and Eiichi Osawa. RoboCup: The robot world cup initiative. In *Proceedings of The First International Conference on Autonomous Agents*. The ACM Press, 1997.

[22] J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33:159–174, 1977.

[23] R. S. Lockhart. *Introduction to Statistics and Data Analysis for the Behavioral Sciences*. Freeman, 1998.

[24] R. M. Neal. Assessing relevance determination methods using DELVE. In C. M. Bishop, editor, *Neural Networks and Machine Learning*, pages 97–129. Springer Verlag, 1998. See also `http://www.cs.utoronto.ca/~delve/`.

[25] U. Nehmzow, M. Recce, and D. Bisset. Towards intelligent mobile robots - scientific methods in mobile robotics. Technical Report UMCS-97-9-1, University of Manchester Computer Science, 1997. Edited collection of papers, see also related special issue of *Journal of Robotics and Autonomous Systems*, in preperation.

[26] David L. Parnas. Software aspects of strategic defense systems. *American Scientist*, 73(5):432–440, 1985. revised version of UVic Report No. DCS-47-IR.

[27] Phoebe Sengers. Do the thing right: An architecture for action expression. In Katia P Sycara and Michael Wooldridge, editors, *Proceedings of the Second International Conference on Autonomous Agents*, pages 24–31. ACM Press, 1998.

[28] Aaron Sloman and Brian Logan. Building cognitively rich agents using the Sim_agent toolkit. *Communications of the Association of Computing Machinery*, 42(3):71–77, March 1999.

[29] L. A. Stein. Challenging the computational metaphor: Implications for how we think. *Cybernetics and Systems*, 30(6):473–507, 1999.

[30] Kristinn R. Thórisson. *Communicative Humanoids: A Computational Model of Psychosocial Dialogue Skills*. PhD thesis, MIT Media Laboratory, September 1996.

[31] Toby Tyrrell. *Computational Mechanisms for Action Selection*. PhD thesis, University of Edinburgh, 1993. Centre for Cognitive Science.

PART II
RESEARCH PAPERS

4. MOBILITY AND BENCHMARKING

Features of Intelligence Required by Unmanned Ground Vehicles

James S. Albus
National Institute of Standards and Technology
Gaithersburg, MD 20899

ABSTRACT

A definition of intelligence is given in terms of performance that can be quantitatively measured. Behaviors required of unmanned ground vehicles are described and computational requirements for intelligent control at seven hierarchical levels in a military scout platoon are outlined. Metrics and measurements are suggested for evaluating the performance of unmanned ground vehicles. Calibrated data and test facilities are suggested to facilitate the development of intelligent systems.

KEYWORDS: intelligence, intelligent systems, unmanned ground vehicles, scout platoon, autonomous vehicles, metrics, measures

1. DEFINITIONS

The definition of intelligence is a controversial subject. Hardly any two persons define intelligence the same. Some even question whether intelligence can be defined at all. Yet, if we are to perform serious research on intelligent systems, we must not only be able to define intelligence, we must be able to quantitatively measure it. Thus, for the purpose of discussion of the issues addressed in this paper, we will define intelligence as follows [1]:

Df: **intelligence**

the ability to act appropriately in an uncertain environment

Df: **appropriate action**

that which maximizes the probability of success

Df: **success**

the achievement or maintenance of behavioral goals

Df: **behavioral goal**

a desired state of the environment that a behavior is designed to achieve or maintain

This definition of intelligence addresses both biological and machine embodiments. It admits a broad spectrum of behaviors, from the simple to the complex. We deliberately do not define intelligence in binary terms (i.e., this machine is intelligent and this one is not, or this species is intelligent and this one is not) and we do not limit our definition of intelligence to behavior that is beyond our understanding. Our definition includes the entire spectrum of intellectual capabilities from that of a paramecium to that of an Einstein, from that of a thermostat to that of the most sophisticated computer system. We include the ability of a robot to spot-weld an automobile body, the ability of a bee to navigate in a field of wild flowers, a squirrel to jump from limb to limb, a duck to land in a high wind, and a swallow to catch insects in flight above a field of wild flowers. We include the ability of blue jays to battle in the bushes for a nesting site, a pride of lions to conduct a coordinated attack on a wildebeest, and a flock of geese to migrate south for the winter. We include a human's ability to bake a cake, play the violin, read a book, write a poem, fight a war, or invent a computer.

Our definition of intelligence recognizes degrees, or levels, of intelligence. These are determined by the following parameters: 1) the computational power and memory capacity of the system's brain (or computer), 2) the sophistication of the processes the system employs for sensory processing, world modeling, behavior generation, value judgment, and communication, and 3) the quality and quantity of information and values the system has stored in its memory. The measure of intelligence is success in solving problems, anticipating the future, and acting so as to maximize the likelihood of achieving goals. Success can be measured by various criteria of performance (including life or death, pain or pleasure, reliability in goal achievement, cost in time and resources, and others.) Different levels of intelligence produce different probabilities of success.

Our definition of intelligence also has many dimensions. For example, the ability to understand what is visually perceived is qualitatively different from the ability to comprehend what is spoken. The ability to reason about mathematics and logic lies along a different dimension from the ability to compose music and verse. The ability to choose wisely involves both the ability to predict the future and the ability to accurately assess the cost or benefit of predicted future states. Along each of these dimensions, there exists a continuum. Thus, the space of intelligent systems is a multidimensional continuum wherein non-intelligent systems occupy a point at the origin.

At a minimum, intelligence requires the ability to sense the environment, to make decisions, and to control action. Higher levels of intelligence may include the ability to recognize objects and events, to represent knowledge in a world model, and to reason about and plan for the future. In advanced forms, intelligence provides the capacity to predict the future, to perceive and understand what is going on in the world, to choose wisely, and to act successfully under a large variety of circumstances so as to survive, prosper, and replicate in a complex, competitive, and often hostile environment.

From the viewpoint of control theory, intelligence might be defined as a knowledgeable "helmsman of behavior." Intelligence is a phenomenon which emerges as a result of the integration of knowledge and feedback into a sensory-interactive, goal-directed control system that can make plans and generate effective purposeful action to achieve goals.

From the viewpoint of psychology or biology, intelligence might be defined as a behavioral strategy that gives each individual a means for maximizing the likelihood of success in achieving its goals in an uncertain and often hostile environment. Intelligence results from the integration of perception, reason, emotion, and behavior in a sensing, perceiving, knowing, feeling, caring, planning, and acting system that can formulate and achieve goals.

2. REQUIREMENTS FOR UNMANNED GROUND VEHICLES

The features of intelligence required by an Unmanned Ground Vehicle (UGV) depends on many factors, such as:

What does the UGV have to do?

Does it simply wander through a lab looking for soft drink cans?

Does it have to operate outside? Travel long distances? Perform difficult tasks?

How complex and uncertain is the environment?

Where is it expected to operate? On well marked roads? On unmarked roads? Gravel or dirt roads? Roads grown up with weeds and brush? Off roads? In tall grass and weeds? In woods? Does it have to cross streams? Are there bridges or fords available? What kind of maps are available? How accurate are they? How recent?

How dynamic and hostile is the environment?

Are there moving obstacles? What are the lighting conditions? Are obstacles located above or below ground level? Are there other agents competing for the goal? Are there enemy agents with deadly weapons?

What are costs, risks, and benefits?

What are the stakes? Life or death? Win or lose?

What are goals?

Attack? Defend? Escape? Detect and track enemy targets? Remain undetected?

What are tasks?

Pick up an object? Use a tool? Dig a ditch? Cross a stream? Establish an observation post? Discover an enemy vehicle? Analyze enemy behavior? Identify a face in a crowd?

What sensors are available?

CCD cameras? FLIRs? LADARs? Radars? Sonars? Inertial? GPS? Beacons? Reflectors? Tactile? Force? Encoders?

What actuators are to be controlled?

Manipulators? Grippers? Power train? Legs or Wheels? Steering? Brakes? Switches?

How much is known apriori?

Maps? Lists of objects and their attributes? State of objects? Behavior of objects? Rules?

What skills and abilities are required?

Locomotion? Manipulation? Perception? Communication? Reasoning? Speech understanding? Written text understanding? In what languages?

The above questions are so open ended that it is futile to try to address all these issues simultaneously. To focus our efforts, we select an example of a problem that is difficult enough to be challenging, well defined enough to quantitatively measure performance, easy enough that it probably can be achieved using available technology, and useful enough that it is worth spending time and resources to solve it. The problem that we have selected it that of an unmanned ground vehicle for military scout operations.

3. A SCOUT PLATOON EXAMPLE

To illustrate the types of issues that will be addressed, an example is given below of a seven level hierarchy for a scout platoon attached to a battalion. The specific numbers and functions described in this example are illustrative only. They are meant only to illustrate how the generic structure and function of an intelligent system might be instantiated in the 4D/RCS architecture [2] designed for the Army's Demo III experimental unmanned ground vehicle program. [3]

Exact numbers for the actual system are still under development.

Level 7 -- Battalion

An armored battalion is a unit that consists of a group of M1 or Bradley companies and a scout platoon. A computational node at level 7 of the 4D/RCS architecture corresponds to a battalion headquarters unit, consisting of a battalion commander, several company commanders, a scout platoon leader, and support staff. (In principle, any or all of these could be humans or intelligent agent software processes. In practice, they are all humans.)

The battalion headquarters unit plans activities and allocates resources for the armored companies and the scout platoon attached to the battalion. Incoming orders to the battalion are decomposed by the battalion commander into assignments for the companies and the scout platoon. Resources and assets are allocated to each subordinate unit, and a schedule is generated for each unit to maneuver and carry out assigned operations. Together, these assignments, allocations, and schedules comprise a plan. The plan may be devised by the battalion commander alone, or in consultation with his subordinate unit leaders. The battalion level planning process may consider the exposure of each unit's movements to enemy observation, and the traversability of roads and cross-country routes. The battalion commander typically defines the rules of engagement for the units under his command and works with his unit leaders to develop a schedule that meets the objectives of the mission orders given to the battalion. In the 4-D/RCS battalion node, plans are computed for a period of about 24 hours(h) and recomputed at least once every 2 h, or more often if necessary. Desired positions for each of the subordinate units at about 2 h intervals are computed.

The 4D/RCS architecture provides a surrogate battalion node in each individual vehicle to perform the functions of the battalion headquarters unit when the vehicle is not in direct communication with its chain of command. The surrogate node plans activities for the vehicle on a battalion level time scale and estimates what platoon and section level operations should be executed to follow that plan. The surrogate battalion node considers the exposure of scout platoon operations to enemy observations, and the traversability of roads and cross-country routes.

In the surrogate battalion node in each vehicle, the 4-D/RCS world model maintains a knowledge database containing a copy of the battalion level knowledge database that is relevant to that vehicle. It contains names and attributes of friendly and enemy forces and of the force levels required to engage them. Maps have a range of 1000 km (i.e. more than the distance that a vehicle is likely to travel in a 24 h day at a Demo III speed of 36 km per hour (10 m/s)) with a resolution of about 400 m. Maps describe the terrain and location of friendly and enemy forces (to the extent that they are known), and roads, bridges, towns, and obstacles such as mountains, rivers, and woods. Battalion level maps may be updated from intelligence reports.

4-D/RCS sensory processing in the surrogate battalion node integrates information about the movement of forces, the level of supplies, and the operational status of all the units in the battalion, plus intelligence about enemy units in the area of concern to the company. This information is used to update maps and lists in the knowledge database so as to keep it accurate and current.

The surrogate battalion node also contains value judgment functions (e.g., calculating the risk of casualties) that enable the battalion commander to evaluate the cost and benefit of various tactical options. To the extent that the knowledge, skills, and abilities in the surrogate battalion node is identical with that in the real battalion node, the surrogate battalion node will make the same decisions as the real battalion headquarters node.

An operator interface allows human operators (either on-site or remotely) to visualize information such as the deployment and movement of forces, the availability of ammunition, and the overall situation within the scope of attention of the battalion commander. The operator can intervene to change priorities, alter tactics, or redirect the allocation of resources.

Output from the battalion level through the company commanders and scout platoon leader comprise input commands to the company/platoon level. Armor company commanders and the scout platoon leader are expected to issue commands to their respective units, monitor how well their units are following the battalion plan, and make adjustments as necessary to keep on plan. New output commands may be issued at any time, and typically consist of tasks expected to require about 2 h to complete.

Level 6—Platoon

A scout platoon is a unit that typically consists of ten HMMWVs or Bradley vehicles organized into one or more sections. For the Demo III project, a scout platoon will consist of six manned HMMWVs and four UGVs. A 4-D/RCS node at the Platoon level corresponds to a scout platoon headquarters unit. It consists of a platoon commander plus his/her section leaders. (Any of these could be humans or intelligent agent software processes, in any combination.) The platoon commander and section leaders plan activities and allocate resources for the sections in the platoon. Platoon orders are decomposed into job assignments for each section. Resources are allocated, and a schedule of activities is generated for each section. Movements are planned relative to major terrain features and other sections within the platoon. Inter-section formations are selected on the basis of tactical goals, stealth requirements, and other priorities. At the platoon level, plans are computed for a period of about 2 h into the future, and replanning is done about every 10 min, or more often if necessary. Section waypoints about 10 min apart are computed.

The surrogate platoon node in each vehicle performs the functions of the platoon headquarters unit when the vehicle is not in direct communication with the chain of command. It plans activities for the vehicle on a platoon level time scale and estimates what vehicle level maneuvers should be executed in order to follow that plan. Movements are planned relative to major terrain features and other vehicles within the platoon.

At the platoon level, the 4-D/RCS world model symbolic database contains names and attributes of targets, and the weapons and ammunition necessary to attack them. Maps with a range of about 100 km (i.e. more than the distance a platoon is likely to travel in 2 h) and resolution of about 40 m describe the location of objectives, and routing between them. Sensory processing integrates intelligence about the location and status of friendly and enemy forces. Value judgment evaluates tactical options for achieving section objectives. An operator interface allows human operators to visualize the status of operations and the movement of vehicles within the section formation. Operators can intervene to change priorities and reorder the plan of operations. Section leaders are expected to sequence commands to their respective sections, monitor how well their sections are following the platoon plan, and make adjustments as necessary to keep on plan. The output from the platoon level through the section leaders are commands issued to sections to perform maneuvers and engage enemy units in particular sectors of the battlefield. Output commands may be issued at any time, but typically are planned to change only about once every 5 min.

Level 5—Section

A scout section is a unit that consists of a group of individual scout vehicles such as HMMWVs and UGVs. A 4-D/RCS node at the section level corresponds to a section leader and vehicle commanders (humans or intelligent software agents). The section leader assigns duties to the vehicles in his section and coordinates the vehicle commanders in scheduling cooperative activities of the vehicles within a section. Orders are decomposed into assignments for each vehicle, and a schedule is developed for each vehicle to maneuver in formation within assigned corridors taking advantage of local terrain features and avoiding obstacles. Plans are developed to conduct coordinated maneuvers and to perform reconnaissance, surveillance, or target acquisition functions. At the section level, plans are computed for about 10 min into the future, and replanning is done about every 1 min, or more often if necessary. Vehicle waypoints about 1 min apart are computed.

The surrogate section node in each UGV performs the functions of the section command unit when the UGV is not in direct communication with the section commander. The surrogate node plans activities for the UGV on a section level time scale and estimates what vehicle level maneuvers should be executed in order to follow that plan.

At the section level, the 4-D/RCS world model symbolic database contains names, coordinates, and other attributes of other vehicles within the section, other sections, and potential enemy targets. Maps with a range of about 10 km and a resolution of about 30 m are typical. Maps at the section level describe the location of vehicles, targets, landmarks, and local terrain features such as buildings, roads, woods, fields, streams, fences, ponds, etc. Sensory processing determines the position of landmarks and terrain features, and tracks the motion of groups of vehicles and targets. Value judgment evaluates plans and computes cost, risk, and payoff of various alternatives. An operator interface allows human operators to visualize the status of the battlefield within the scope of the section, or to intervene to change priorities and reorder the sequence of operations or selection of targets. Vehicle commanders issue commands to their respective vehicles, monitor how well plans are being followed, and make adjustments as necessary to keep on plan. Output commands to individual vehicles to engage targets or maneuver relative to landmarks or other vehicles may be issued at any time, but on average are planned for tasks that last about 1 min.

Level 4—Individual vehicle

The vehicle is a unit that consists of a group of subsystems, such as locomotion, attention, communication, and mission package. A manned scout vehicle may have a driver, vehicle commander, and a lookout. Thus, a 4-D/RCS node at the vehicle level corresponds to a vehicle commander plus subsystem planners and executors. The vehicle commander assigns jobs to subsystems and schedules the activities of all the subsystems within the vehicle. A schedule of waypoints is developed by the locomotion subsystem to avoid obstacles, maintain position relative to nearby vehicles, and achieve desired vehicle heading and speed along the desired path on roads or cross-country. A schedule of tracking activities is generated for the attention subsystem to track obstacles, other vehicles, and targets. A schedule of activities is generated for the mission package and the communication subsystems. Waypoints and task activities about 5 s apart out to a planning horizon of 1 min are replanned every 5 s, or more often if necessary.

At the vehicle level, the world model symbolic database contains names (identifiers) and attributes of objects -- for example, the size, shape, and surface characteristics of roads, ground cover, or objects such as rocks, trees, bushes, mud, and water. Maps are generated from on-board sensors with a range of about 500 m and resolution of 4 meters. These maps are registered and overlaid with 40 meter resolution data from Section level maps. Maps represent object positions (relative to the vehicle) and dimensions of road surfaces, buildings, trees, craters, and ditches. Sensory processing measures object dimensions and distances, and computes relative motion. Value judgment evaluates trajectory planning and sensor dwell time sequences. An operator interface allows a human operator to visualize the status of operations of the vehicle, and to intervene to change priorities or steer the vehicle through difficult situations. Subsystem controller executors sequence commands to subsystems, monitor how well plans are being followed and modify parameters as necessary to keep on plan. Output commands to subsystems may be issued at any time, but typically are planned to change only about once every 5 s.

Level 3—Subsystem level

Each subsystem node is a unit consisting of a controller for a group of related Primitive level systems such as Primitive mobility, Gaze control, Communication, and Mission package sub-subsystems. A 4-D/RCS node at the Subsystem Level assigns jobs to each of its Primitive sub-subsystems and coordinates the activities among them. A schedule of Primitive mobility waypoints and Primitive mobility actions is developed to avoid obstacles. A schedule of pointing commands is generated for aiming cameras and sensors. A schedule of messages is generated for communications, and a schedule of actions is developed for operating the mission package sub-subsystems. The Primitive mobility way points are about 500 ms apart out to a planning horizon of about 5 s in the future. A new plan is generated about every 500 ms.

At the Subsystem level, the world model symbolic database contains names and attributes of environmental features such as road edges, holes, obstacles, ditches, and targets. Vehicle centered maps with a range of 50 meters and resolution of 40 cm are generated using data from range sensors. These maps represent the shape and location of terrain features and obstacle boundaries. The Demo III LADAR and stereo cameras measure position and range (out to about 50 m) of surfaces in the environment. Sensory processing computes surface properties such as dimensions, area, orientation, texture, and motion. Value judgment supports planning of steering and aiming computations, and evaluates sensor data quality. An operator interface allows a human operator to visualize the state of the vehicle, or to intervene to change mode or interrupt the sequence of operations. Subsystem executors compute at a 5 Hz clock rate. They sequence commands to primitive systems, monitor how well plans are being followed, and modify parameters as necessary to keep on plan. Output commands to Primitive sub-subsystems may be issued at any 200 ms interval, but typically are planned to change on average about once every 500 ms.

Level 2— Primitive level

Each node at the primitive level is a unit consisting of a group of controllers that plan and execute velocities and accelerations to optimize dynamic performance of components such as steering, braking, acceleration, gear shift, camera pointing, and weapon loading and pointing, taking into consideration dynamical interaction between mass, stiffness, force, and time. Communication messages are encoded into words and strings of symbols. Velocity and acceleration set points are planned every 50 ms out to a planning horizon of 500 ms.

The world model symbolic database contains names and attributes of state variables and features such as target trajectories and edges of objects. Maps are generated from camera data. Five meter maps have a resolution of about 4 cm. Driving plans can be represented by predicted tire tracks on the map, and visual attention plans by predicted fixation points in the visual field.

Sensory processing computes linear image features such as occluding edges, boundaries, and vertices and detects strings of events. Value judgment cost functions support dynamic trajectory optimization. An operator interface allows a human operator to visualize the state of each controller, and to intervene to change mode or override velocities. Primitive level executors keep track of how well plans are being followed, and modify parameters as necessary to keep within tolerance. Primitive executors compute at a 20 Hz clock rate. Output commands are issued to the Servo level to adjust set points for vehicle steering, velocity, and acceleration or for pointing sensors or weapons platforms. Output commands are issued every 50 ms.

Level 1—Servo level

Each node at the servo level is a unit consisting of a group of controllers that plan and execute actuator motions and forces, and generate discrete outputs. Communication message bit streams are produced. The servo level transforms commands from component to actuator coordinates and computes motion or torque commands for each actuator. Desired forces, velocities, and discrete outputs are planned for 20 ms intervals out to a planning horizon of 50 ms.

The world model symbolic database contains values of state variables such as actuator positions, velocities, and forces, pressure sensor readings, position of switches, and gear shift settings. Sensory processing detects events, and scales and filters data from individual sensors that measure position, velocity, force, torque, and pressure. Sensory processing also computes pixel attributes in images such as spatial and temporal gradients, stereo disparity, range, color, and image flow. An operator interface allows a human operator to visualize the state of the machine, or to intervene to change mode, set switches, or jog individual actuators. Executors servo actuators and motors to follow planned trajectories. Position, velocity, or force servoing may be implemented, and in various combinations. Servo executors compute at a 200 Hz clock rate. Motion output commands to power amplifiers specify desired actuator torque or power every 5 ms. Discrete output commands produce switch closures and activate relays and solenoids.

The above example illustrates how the 4-D/RCS multilevel hierarchical architecture assigns different responsibilities and duties to various levels of the hierarchy with different range and resolution in time and space at each level. At each level, sensory data is processed, entities are recognized, world model representations are maintained, and tasks are decomposed into parallel and sequential subtasks, to be performed by cooperating sets of agents. At each level, feedback from sensors reactively closes a control loop allowing each agent to respond and react to unexpected events.

At each level, there is a characteristic range and resolution in space and time, a characteristic bandwidth and response time, and a characteristic planning horizon and level of detail in plans. The 4-D/RCS architecture thus organizes the planning of behavior, the control of action, and the focusing of computational resources such that functional processes at each level have a limited amount of responsibility and a manageable level of complexity.

4. DEMO III CONTROL HIERARCHY

Figure 1 is a high-level block diagram of the first five levels in the 4-D/RCS architecture for Demo III. On the right, Behavior Generation modules decompose high level mission commands into low level actions. The text beside the Planner and Executor at each level indicates the planning horizon, replanning rate, and reaction latency, and the rate at which new commands are typically generated at each level. Each planner has a world model simulator that is appropriate for the problems encountered at its level.

In the center of Figure 1, each map as a range and resolution that is appropriate for path planning at its level. At each level, there are symbolic data structures and segmented images with labeled regions that describe entities, events, and situations that are relevant to decisions that must be made at that level. On the left is a sensory processing hierarchy that extracts information from the sensory data stream that is needed to keep the world model knowledge database current and accurate.

At the bottom of Figure 1 are actuators that act on the world and sensors that measure phenomena in the world. The Demo III vehicles will have a variety of sensors including a laser range imager (LADAR), stereo CCD (charge coupled device) cameras, stereo forward looking infra red (FLIR) devices, a color CCD, a vegetation penetrating radar, GPS (Global Positioning System), an inertial navigation package, actuator feedback sensors, and a variety of internal sensors for measuring parameters such as engine temperature, speed, vibration, oil pressure, and fuel level. The vehicle also will carry a Reconnaissance, Surveillance, and Target Acquisition (RSTA) mission package that will include long-range cameras and FLIRs, a laser range finder, and an acoustic package.

In Figure 1, the bottom (Servo) level has no map representation. The Servo level deals with actuator dynamics and reacts to sensory feedback from actuator sensors. The Primitive level map has range of 5 m with resolution of 4 cm. This enables the vehicle to make small path corrections to avoid bumps and ruts during the 500 ms planning horizon of the Primitive level. The Primitive level also uses accelerometer data to control vehicle dynamics and prevent roll-over during high speed driving.

The Subsystem level map has range of 50 m with resolution of 40 cm. This map is used to plan about 5 s into the future to find a path that avoids obstacles and provides a smooth and efficient ride. The Vehicle level map has a range of 500 m with resolution of 4 m. This map is used to plan paths about 1 min into the future taking into account terrain features such as roads, bushes, gullies, or tree lines. The Section level map has a range of 5000 m with resolution of about 40 m. This map is used to plan about 10 m into the future to accomplish tactical behaviors. Higher level maps (not shown in Figure 1) are used to plan section and platoon missions lasting about 2 and 24 h respcctively. These are derived from military maps and intelligence provided by the digital battlefield database.

4D/RCS planners are designed to generate new plans well before current plans become obsolete. Thus, action can always take place in the context of a recent plan, and feedback through the executors can close reactive control loops using recently selected control parameters. To meet the demands of Demo III, the 4D/RCS architecture specifies that replanning should occur within about one-tenth of the planning horizon at each level (e.g., replanning at the Vehicle level will occur about every 5 s.)

Executors can react to sensory feedback even faster (e.g., reaction at the Vehicle level will occur within 500 ms). If the Executor senses an error between its output CommandGoal and the predicted state (status from the subordinate BG Planner) at the GoalTime, it may react by modifying the commanded action so as to cope with that error. This closes a feedback loop through the Executor at that level within the specified reaction latency.

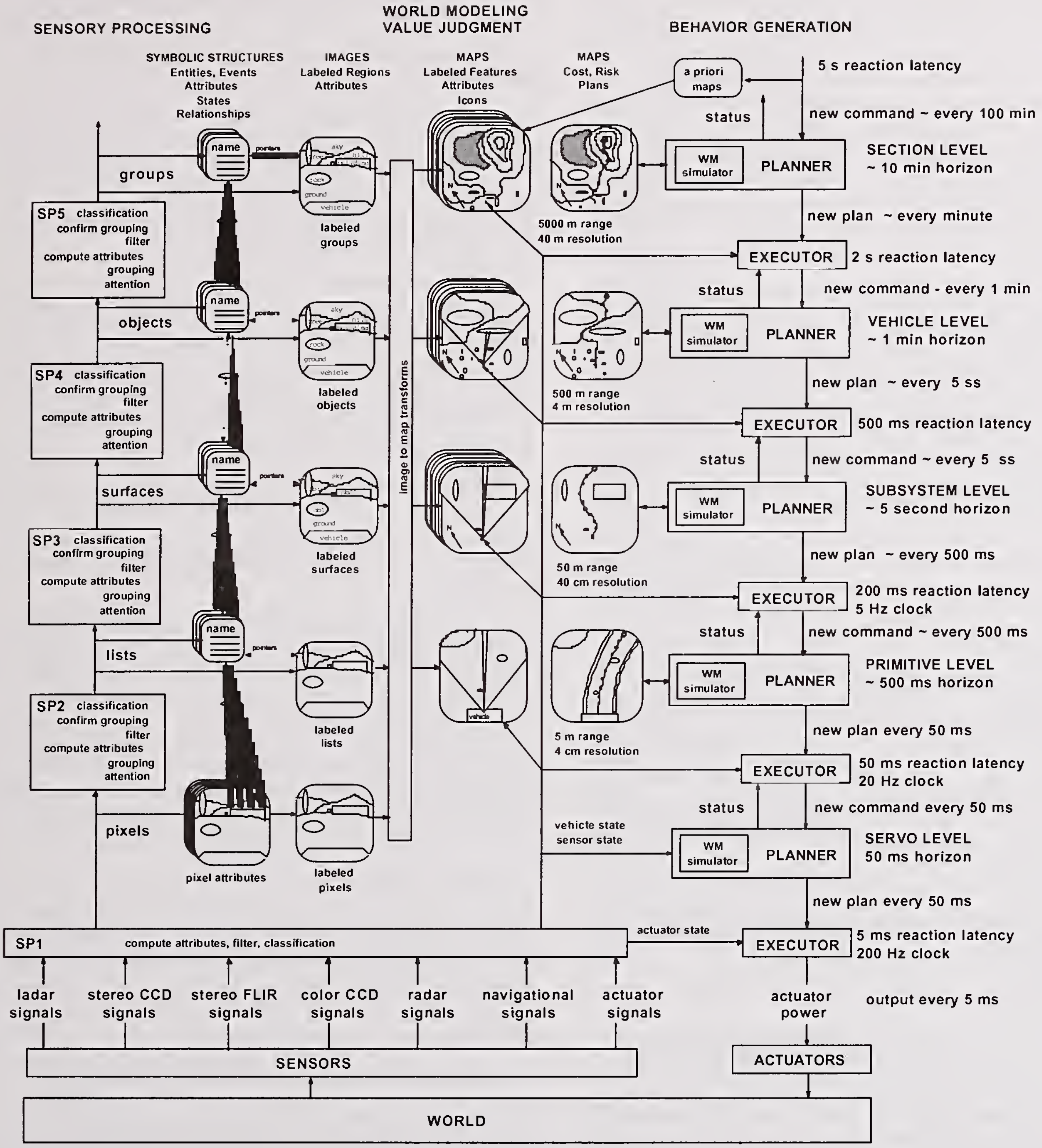


Figure 1. Five levels of the 4-D/RCS architecture. On the right are Planner and Executor modules. In the center are maps for representing terrain features, road, bridges, vehicles, friendly/enemy positions, and the cost and risk of traversing various regions. On the left are Sensory Processing functions, symbolic representations of entities and events, and segmented images with labeled regions.

The type of Executor reaction depends on the size and nature of the detected error. If the error is small, the Executor may simply modify its CommandedAction in a manner designed to reduce the error. For example, if the status reported from the subordinate planner indicates that the vehicle is going to arrive at the goal point late, the Executor might modify its CommandedAction to speed up or delete some low priority activities. However, if the error is out of range, the Executor may select a stored emergency plan from an exception handler, substitute it for the current plan, and modify its CommandedAction and CommandGoal to its subordinate planner appropriately. For example, an event such as the discovery of an unexpected obstacle in the AM planned path (generated by the Vehicle Planner) may cause the AM planner to make a plan that deviates significantly from its commanded goal. In this case, the Vehicle level Executor may modify its CommandedAction in a manner designed to buy time for the Vehicle level Planner to generate a new AM plan. For example, it may command the AM level to reduce speed or stop and direct AM driving cameras or RSTA sensors to collect information about the obstacle while a new AM plan is being generated by the Vehicle level planner. All of this Executor response should take place within the 500 ms reaction latency of the Vehicle level Executor.

Typically, evoking an emergency plan will cause the Executor to request its Planner to immediately begin a new replanning cycle. As shown in Figure 1, the period required for replanning at the Vehicle level is 5 s. The replanning period at the AM level is 0.5 s. Thus, the emergency plan evoked by the Vehicle level Executor can handle the problem of what the AM level should plan to do over the next 5 s while the Vehicle level planner generates a new AM plan out to its 1 min planning horizon.

5. GENERIC BEHAVIORS OF SCOUT VEHICLES

Navigate from A to B

Point A may be several km from point B. What kind of roads are available? How much traffic will be present? A scout vehicle may be required to stay off of roads, to maneuver through hilly fields and woods, and cope with fences, washes, and streams.

Avoid obstacles

The simplest obstacles are those that stick up from flat ground and are not obscured by foliage. The most difficult are ditches that are obscured by foliage. It is important to be able to distinguish grass and weeds that the vehicle can drive through from grass and weeds that conceal obstacles. In some cases, the only way to tell the difference is to drive slowly and stop when the vehicle encounters stiff resistance, or when the front wheels drop over the edge of a ditch, or sink into the mud.

Compute terrain attributes and classify terrain features

The first requirement is to map the terrain geometry and topology. The second is compute attributes such as color, texture, slope, size, and shape of regions of terrain. The third is to compare attributes of terrain regions with class attributes so as to classify terrain regions as road, dirt, grass, rocks, brush, trees, and bogs.

Drive autonomously

Driving autonomously covers a wide range of situations. Driving on an empty freeway is quite different from driving in downtown Istanbul. Driving with traffic on a freeway requires the ability to recognize lane markings, detect and track other vehicles, detect and avoid obstacles in the roadway, and obey road signs.

Driving at normal human speeds on narrow roads and cross country is more difficult. Road edges may be poorly defined and lane markings often do not exist. There may be bumps or ditches that will damage the vehicle if struck at high speeds.

Autonomous driving in suburban or downtown streets requires the ability to detect and predict the behavior of pedestrians, other vehicles, to read road signs, and respond to traffic signals, including hand signals from humans.

In driving cross country, there is no guarantee that a chosen path is even feasible. There may be hidden obstacles such as ditches, streams, fences, hills, brush, or woods that are impassable. The vehicle must be able to back up, and try alternate routes when the planned path is blocked.

Classify landmarks, objects, places, and situations

It is easy to get lost. GPS is not always available. Critical path waypoints may not appear on a map, or may be incorrectly represented. The unexpected appearance of an enemy may require immediate action. The ability to recognize a likely spot for an enemy sniper in time to take evasive action may be critical to survival.

Recognize and track other vehicles, avoid collisions

On-coming traffic on narrow roads is a major problem. One must drive very close to oncoming vehicles to stay on the road. One must estimate whether the oncoming vehicle is in its own lane on its own side of the road, and whether there is room on the road for two vehicles to safely pass. To do that one must detect the road edges at a great distance and measure the relative position of the on-coming vehicle between the road edges. There is very little margin for error in space or time.

Predict behavior of pedestrians and other vehicles in traffic

Driving in traffic requires the self vehicle to not only detect, but to predict where pedestrians and other vehicles will be in the future. For example, on a two lane road, on-coming traffic may consist of one vehicle passing another. The self vehicle must predict whether the on-coming vehicle in the self vehicle lane will return to its own side before a head-on collision occurs. On a one lane road, it may be necessary for the self vehicle to pull over and let an on-coming vehicle pass, or wait for the on-coming vehicle to pull over so that the self vehicle can pass. On a narrow mountain road, it may be necessary to back up to a place where it is wide enough for two vehicles to pass each other.

Learn from experience and from human instructors

Adjust behavior to situation and priorities. Use reward and punishment from human instructors to learn skills and behaviors. Use experience from multiple simulated scenarios to learn from experience.

6. METRICS AND MEASURES

A metric is a unit of measure. Examples include the meter, the second, the kilogram, the volt, Plank's constant, and Avogadro's number.

Measurements are made by comparing something against the unit of measure. A measurement can be made of the length of the coastline of the British Isles, the height of the Eiffel Tower, the mass of the Queen Mary, the length of a day, or the charge on an electron. There are many parameters related to measurement including accuracy, precision, resolution, observability, and uncertainty.

What is it about intelligent systems that can be measured? If an intelligent system is defined as a system with the ability to act appropriately in an uncertain environment, then we can measure the appropriateness of its behavior. And, if appropriate behavior is defined as that which increases the likelihood of achieving a goal, then the ability of a system to achieve goals in an uncertain environment is a measure of intelligence.

At least three things are required to measure the ability of a system to achieve goals. First, we need to define the goals and set criteria for achieving them. Second, we need to provide an environment in which to make the measurements. Third, we need to define a procedure for scoring performance that takes into account the difficulty of the goals, and the complexity and uncertainty of the environment

What kinds of measurements can be used to measure performance? One possibility is to develop one or more benchmark tests, and measure speed, accuracy, efficiency, level of difficulty, and cost. These measurements can then be weighted for importance and summed to provide an overall score.

Another approach is to devise competitions wherein different intelligent systems can compete against each other for a score. Competitions can involve direct physical interactions such as in football or tennis, measurements of time as in skiing or bobsleding, or competitions that consider both style and difficulty as in ice skating, diving, and gymnastics. Again, performance measurements can be weighted for importance and summed to provide a score.

What kind of metric can be used to measure the performance of an intelligent systems? One possible metric is the performance of a human being. Another possible metric is the performance of a standard baseline system. In either case, the performance of the intelligent system under test can be compared with the performance of a human being (or baseline system) under similar conditions. The difference in performance, the level of difficulty of the test, and the weighting for importance of the test all combine to give a score.

Measures of performance can be devised for subsystem performance, individual system performance, or group or team performance. For example, for subsystems, benchmark tests can be devised to measure the performance of sensory processing algorithms, world model predictors, or behavior generation planners. One might measure the difference between predictions and observations, or the difference between plans and actions. Benchmark tests can also be devised to measure the accuracy of knowledge about the world. For example, one can measure the difference between perceived terrain geometry derived from sensors and ground truth from calibrated test courses. One can measure the latency between requesting and receiving information about the world. Individual system performance can be measured and scored against standard tasks that are typically required of human scout vehicles. Similarly, team performance can be measured and scored in war games wherein opposing forces are tested in battle fighting scenarios.

What is needed?

Calibrated test facilities are needed to test the performance of sensors and systems in the field under realistic conditions. High fidelity simulation facilities are needed to generate repeatable test data for software debugging and testing. Data from calibrated sensors, mixed with a known noise, and accompanied by ground truth are needed to test sensory processing and world modeling algorithms. World model data with values assigned to entities and events is needed to test behavior generation planning and control algorithms.

Large scale test and training facilities are needed to test performance of systems in large scale operations and to develop tactics and training for integration of autonomous systems with manned forces. A wide variety of benchmark tests and competitions are needed to test intelligent system performance under a wide variety of environmental conditions. A rigorous regimen of testing, debugging, and reliability engineering will be needed before intelligent systems become robust enough to operate reliably under a wide variety of operational conditions.

6. REFERENCES

[1] Albus, J.S., "Outline for a Theory of Intelligence," IEEE Transactions on Systems, Man and Cybernetics, Vol. 21, No. 3, pgs. 473-509, May/June, 1991

[2] Albus, J. *4D/RCS: A Reference Model Architecture for Demo III, Version 1.0, NISTIR 5994,* Gaithersburg, MD, 1998

[3] Shoemaker, C., Bornstein, J., Myers, S., and Brendle, B. "Demo III: Department of Defense testbed for unmanned ground mobility," *SPIE Conference on Unmanned Ground Vehicle Technology, SPIE Vol. 3693*, Orlando, FA, April, 1999

A Standard Test Course for Urban Search and Rescue Robots

Adam Jacoff, Elena Messina, John Evans
Intelligent Systems Division
National Institute of Standards and Technology
Gaithersburg, MD 20899-8230

ABSTRACT

One approach to measuring the performance of intelligent systems is to develop standardized or reproducible tests. These tests may be in a simulated environment or in a physical test course. The National Institute of Standards and Technology is developing a test course for evaluating the performance of mobile autonomous robots operating in an urban search and rescue mission. The test course is designed to simulate a collapsed building structure at various levels of fidelity. The course will be used in robotic competitions, such as the American Association for Artifical Intelligence (AAAI) Mobile Robot Competition and the RoboCup Rescue. Designed to be highly reconfigurable and to accommodate a variety of sensing and navigation capabilities, this course may serve as a prototype for further development of performance testing environments. The design of the test course brings to light several challenges in evaluating performance of intelligent systems, such as the distinction between "mind" and "body" and the accommodation of high-level interactions between the robot and humans. We discuss the design criteria for the test course and the evaluation methods that are being planned.

KEYWORDS: *performance metrics, autonomous robots, mobile robots, urban search and rescue*

1. INTRODUCTION

The Intelligent Systems Division of the National Institute of Standards and Technology is researching how to measure the performance of intelligent systems. One approach being investigated is the use of test courses for evaluating autonomous mobile robots operating in an urban search and rescue scenario. Urban search and rescue is an excellent candidate for deploying robots, since it is an extremely hazardous task. Urban Search and Rescue (USAR) refers to rescue activities in collapsed building or man-made structures after a catastrophic event, such as an earthquake or a bombing. Japan has an initiative, based on the RoboCup robots, that focuses on multi-agent approaches to the simulation and management of major urban disasters [1]. The real-world utility and manifold complexities inherent in this domain make it attractive as a "challenge" problem for the mobile autonomous robots community. For a description of the issues pertaining to intelligent robots for search and rescue, see [2].

Figures 1 and 2 illustrate the type of environment that a rescuer has to confront with a collapsed building. There is totally unstructured rubble, which may be unstable and contain many hazards. Victims' locations and conditions must be established quickly. Every passing minute reduces the chances of saving a victim.

This type of environment stresses the mobility, sensing, and planning capabilities of autonomous systems. The robots must be able to crawl over rubble, through very narrow openings, climb stairs or ramps, and be aware of the possibility of collapses of building sections. The sensors are confronted with a dense, variable, and very rich set of inputs. The robot has to ascertain how best to navigate through the area, avoiding hazards, such as unstable piles of rubble or holes, yet maximizing the coverage. The robot also has to be able to detect victims and ideally, determine their condition and location. The robot has to make careful decisions, planning its path and strategy, and taking into account the time constraints.

A near-term measure of success for robots in a search and rescue mission would be to scout a structure, map its significant openings, obstacles, and hazards, and locate victims. The robots would communicate with victims, leaving them with an emergency kit that contains a radio, water, and other supplies, and transmit a map, including victim locations and conditions, to human supervisors. Humans would then plan the best means of rescuing the victims, given the augmented situational awareness.

Search and rescue missions are not amenable to teleoperation due to the fact that most of the radio frequencies are reserved by emergency management agencies. Obstructions and occlusions also diminish the effectiveness of radio transmissions. Tethers are not typically practical in the cluttered environment in which these robots must operate.

2. Urban Search and Rescue as a robotic challenge

A search and rescue mission is extremely challenging and dangerous for human experts. This is a highly unstructured and dynamic environment, where the mission is time critical. Very little *a priori* information about the environment or building may exist. If any exists, it will almost certainly be obsolete, due to the collapse.

Urban Search and Rescue is therefore attractive as a mission framework in which to measure intelligence of autonomous robots. The high degree of variability and unpredictability demand high adaptation and sophisticated decision-making skills from the robots. Robots will need to quickly and continually assess the situation, both in terms of their own mobility and of the likelihood of locating more victims. USAR missions are amenable to cooperation, which can be considered another higher-level manifestation of intelligence. We propose that any robot or team of robots that is able to successfully and efficiently carry out USAR missions would be considered intelligent by most standards.

In the following sections, we will briefly discuss how USAR missions tax specific components of an intelligent system.

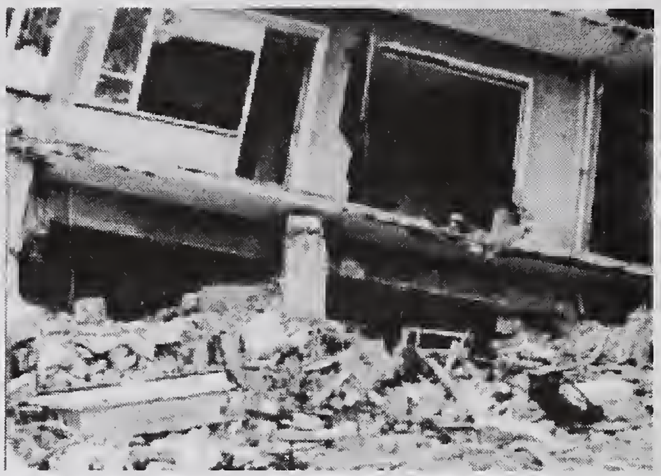


Figure 1: Partially Collapsed Building from Turkey Earthquake

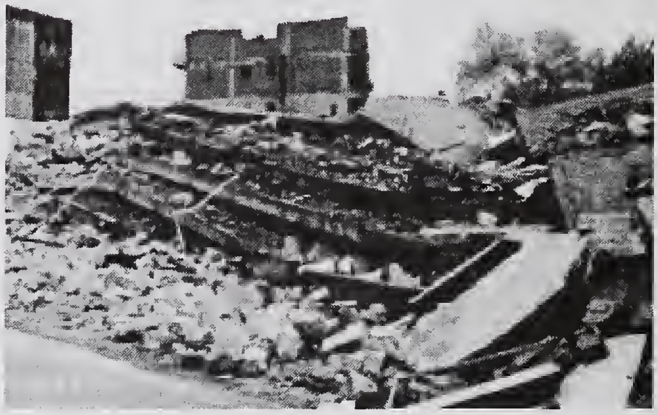


Figure 2: Totally Collapsed Building from Turkey Earthquake

2.1 Mobility

As can be seen from Figures 1 and 2, the mobility requirements for search and rescue robots are challenging. They must be able to crawl over piles of rubble, up and down stairs and steep ramps, through extremely small openings, and take advantage of pipes, tubes, and other unconventional routes. The surfaces that they must traverse may be composed of a variety of materials, including carpeting, concrete blocks, wood, and other construction material. The surfaces may also be highly unstable. The robot may destabilize the area if it is too heavy or if it bumps some of the rubble. There may be gaps, holes, sharp drop-offs, and discontinuities in the surfaces that the robot traverses.

2.2 Sensing

In order to be able to explore an USAR site and successfully navigate in this environment, the robot's sensing and perception must be highly sophisticated. Lighting will be variable and may be altogether missing. Surface geometry and materials may absorb emitted signals, such as acoustic, or they may reflect them. For truly robust perception, the robots should emulate human levels of vision.

The presence of victims may be manifested through a variety of signals. The stimuli that the robots have to be prepared to process include

- Acoustic – victims may be calling out, moaning softly, knocking on walls, or otherwise generating sounds. There will be other noises in the environment due to shifting materials or coming from other USAR entities.
- Thermal – a body will emit a thermal signature. There may be other sources of heat, such as radiators or hot water.
- Visual -- a multiplicity of visual recognition capabilities, based on geometric, color, textural, and motion characteristics, will be exercised. Recognizing human characteristics, such as limbs, color of skin, clothing is important. Motion of humans, such as waving, must be detected. Confusing visual cues may come from wallpaper, upholstery or curtain material, strewn clothing, and moving objects, such as curtains blown by a breeze.

2.3 Knowledge Representation

In order to support the sophisticated planning and decision-making that is required, the robot must be able to leverage a rich knowledge base. This entails both *a priori* expertise or knowledge, such as how to characterize the traversability of a particular area, as well as gained information, such as a map that is built up as it explores. It must develop rich three-dimensional spatial maps that contain areas it or other robots have and haven't yet seen, victim and hazard locations, and potential quick exit routes. The maps from several robots may need to be shared and merged.

A variety of types of knowledge will be required in order to successfully accomplish search and rescue tasks. Higher-level knowledge, which may be symbolic, includes representations of what a "victim" is. This is a multi-facetted definition, which includes the many manifestations that imply a victim's presence.

2.4 Planning

An individual robot must be able to plan how to best cover the areas it has been assigned. The time-critical nature of its work must be taken into account in its planning. It may need to trade off between delving deeper into a structure to find more victims and finding a shortcut back to its human supervisors to report on the victims it has already found.

2.5 Autonomy

As mentioned above, it is not currently practical to assume that the robots will be in constant communication with human supervisors. Therefore, the robots must be able to operate autonomously, making and updating their plans independently. In some circumstances, there may be limited-bandwidth communications available. In this case, the robots may be able to operate under a mixed-initiative mode, where they have high-level interactions with humans. The communications should be akin to those that a human search and rescue worker may have with his or her supervisor. It definitely would not be of a teleoperative nature.

2.6 Collaboration

Search and rescue missions seem ideally suited for deploying multiple robots in order to maximize coverage. An initial strategy for splitting up the area amongst the robots may be devised. Once they start executing this plan, they will revise and adapt their trajectories based on the conditions that they encounter. Information sharing between the robots can improve their efficiency. For example, if a robot detects that a particular passageway that others may need to use is blocked, it would communicate that to its peers. The robots should therefore collaborate and cooperate as they jointly perform the mission. They may be centrally or decentrally controlled. The robots themselves may all have the same capability, or they may be heterogeneous, meaning that they have different characteristics. Heterogeneous robot teams may apply the marsupial approach, where a larger robot transports smaller ones to their work areas and performs a supervisory function.

3. Measuring the Performance of USAR Robots

We have described briefly the requirements for autonomous urban search and rescue robots. We will now discuss approaches to testing their capabilities in achieving a USAR mission.

The approach being taken by the upcoming USAR robot competitions that will use the NIST test course is based on a point system. The goal of the robots is to maximize the number of victims and hazards located, while minimizing the amount of time to do so and the disruption of the test course.

Specifically, the AAAI Mobile Robot competition [1] will use Olympic-style scoring. Each judge will have a certain number of points that can be awarded based on their measuring certain quantitative and qualitative metrics. Robots receive points for

- Number of victims located
- Number of hazards detected
- Mapping of victim and hazard locations
- Staying within time limits
- Dropping off a package to victims representing first aid, a radio, or food and water
- Quality of communications with humans
- Tolerance of communications dropout

They lose points for

- Causing damage to the environment, victims, or themselves (e.g., destabilizing a structure)
- Failing to exit within time limits

In certain sections of the test course, robots are allowed to have high-level communications with humans. These communications must be made visible to the judges. Metrics for evaluating the quality of the communications include "commands" per minute and/or bandwidth used. Fewer commands per minute and less bandwidth per minute receive better scores. Tolerance of communications disruption is an important capability and will be given greater difficulty weighting. A team may request that the judges simulate communication disruptions at any point in order that the robots demonstrate how to recover. Examples of recovery would be to move to a location where there is better chance of communication, making decisions autonomously instead of consulting humans, or utilizing companion robots to relay the information to the humans.

For teams consisting of multiple robots, the advantage of cooperating or interacting robot must be demonstrated. This can be either in performing the task better, or performing the task more economically. Multi-robot teams should have a time speedup that is greater than linear, or may be able to perform the tasks with less overall power consumption or cost. The scoring will factor in the number of robots, types of robots, types or mixture of sensors, etc., in determining the performance of a team.

The RoboCup Rescue competition, sponsored by Robot World Cup Initiative, takes an evaluation benchmarking approach. Initially, there are 3 benchmark tasks. The current tasks are victim search, victim rescue, and a combination of victim search and rescue. Additional ones will be added as the competition and participants evolve. The RoboCup Rescue includes a simulation infrastructure in which teams can compete, as well as the use of the NIST test course.

Their evaluation metrics are still under development. Examples of criteria that have been published on their web site [4] include:

- Recovery rate, expressed as percentage of victims identified versus number under the debris.
- Accuracy rate, computed as the number of correctly identified victims divided by the total number of identified victims.
- Operational loading, which is the number of operations that a human has to perform in order to enable to robots to perform their tasks.
- If rescuing victims, the total time it takes to rescue all victims.
- Total damage caused to victims in attempting to rescue them.

4. The test course design

The test course which NIST designed for the AAAI Mobile Robot Competition was designed with three distinct areas of increasing verisimilitude and difficulty. Overall, the course is meant to represent several of the sensing, navigation, and mapping challenges that exist in a real USAR situation. As discussed above, these are challenges that correlate well with general characteristics desirable in mobile, autonomous robots that may operate in other types of missions. In the design of the course, tradeoffs were made between realism and reproducible and controlled conditions. In order to be able to evaluate the performance of robots in specific skill areas, certain portions of the course may look unrealistic or too simplified. This idealization is necessary in order to abstract the essential elements being exercised, such as a the ability to deal with a particular sensing challenge.

Given the controlled conditions that the test course provides, it is possible to have multiple robots or teams face the identical course and have their performances compared. This should yield valuable information about what approaches to robotic sensing, planning, and world modeling work best under certain circumstances.

The course is highly modular, allowing for reconfiguration before and during a competition. Judges may swap wall panels that are highly reflective for some that are fabric-covered, for example, or victims may be relocated. This reconfigurability can serve to avoid having robot teams “game” the course, i.e., program their robots to have capabilities tailored to the course they’ve seen previously. The reconfigurability can serve to provide more realism as well. A route that the robot used previously may become blocked, forcing the robot to have to find an alternative way.

The three areas of the course are described below. Note that the use of color in the names of the section is for labeling purposes only and does not mean that the courses are primarily colored in their namesake color. A representative schematic of the test course is shown in Figure 3, at the end of this paper.

4.1 Yellow Course

Given the fact that participating teams, at least initially, will primarily be from universities that may not have access to new agile robotic platforms, one design requirement was to have an area within the course where the mobility challenges are minimal. We call this area the “Yellow course.” The floor of the yellow course is flat and of uniform material. Passageways are wide enough to permit large robots, up to about 1 meter diameter, to pass easily.

Yet the Yellow course allows teams with sophisticated perception and planning to exercise their robots’ capabilities. Some sensing challenges are as difficult in this section as in the others. There will be highly reflective and highly absorbent material on walls. Certain wall panels will be clear Plexiglas, whereas others will be covered in brightly patterned wallpaper. Some areas may be dimly lit or accessible only from one direction. Victims will be represented in all modalities (i.e., acoustically, visually, through motion, thermally, etc.) and may be hidden from view under furnishings or in closed areas.

4.2 Orange Course

The Orange course is of intermediate difficulty. A second story is introduced, and there are routes that only smaller robots may pass through. The robots may have to climb stairs or a ramp in order to reach victims. Flooring materials of various kinds, such as carpeting, tile, and rubber, are introduced. Hazards, such as holes in the floor, exist. In order to be effective, the robot will have to plan in a three-dimensional space. Larger robots will be able to navigate through some portions of this course, but not all.

4.3 Red Course

The Red course poses the most realistic representation of a collapsed structure. We do not anticipate that any of the contestants will be able to successfully complete the red course in the first or perhaps even second years. However, this section provides a performance goal for the teams to strive for. In the Red section, piles of rubble abound, lighting is minimal or non-existent, and passageways are very narrow. The course is highly three-dimensional, from a mapping perspective. Not only are there two floors, but the rubble piles that the robot has to traverse may need to be mapped as well. Passageways under the rubble or through pipes may have to be used by the robots to reach certain areas or to get closer to victims. There are some portions of this course that can be traversed by the larger class of robots, but they would not be able to reach most of the victims. Larger robots would be best suited in marsupial configurations in this area.

5. Conclusion

An Urban Search and Rescue application for autonomous mobile robots poses several challenges that can be met only by highly intelligent systems. The variability, risk, and urgency inherent in USAR missions makes this a good framework in which to begin measuring performance in controlled and reproducible situations. We believe that the test course we are developing can serve to elucidate performance measures for overall systems, as well as for components of intelligent systems.

6. References

[1] http://www.aic.nrl.navy.mil/~schultz/aaai2000/menu-bar-vert.html

[2] Casper, J., and Murphy, R.R., "Issues in Intelligent Robots for Search and Rescue," SPIE Ground Vehicle Technology II, Orlando, FL, April 2000.

[3] Kitano, H., et al., 'RoboCup Rescue: Search and Rescue in Large-Scale Disasters as a Domain for Autonomous Agents Research," Proceedings of the IEEE Conference on Man, Systems, and Cybernetics, 1999.

[4] http://www.robocup.org/games/36.html

Acknowledgements

The development of the USAR test course was carried out with the support from the Defense Advanced Research Agency (DARPA) MobileAutonomous Robot Software Program and the Naval Research Laboratory.

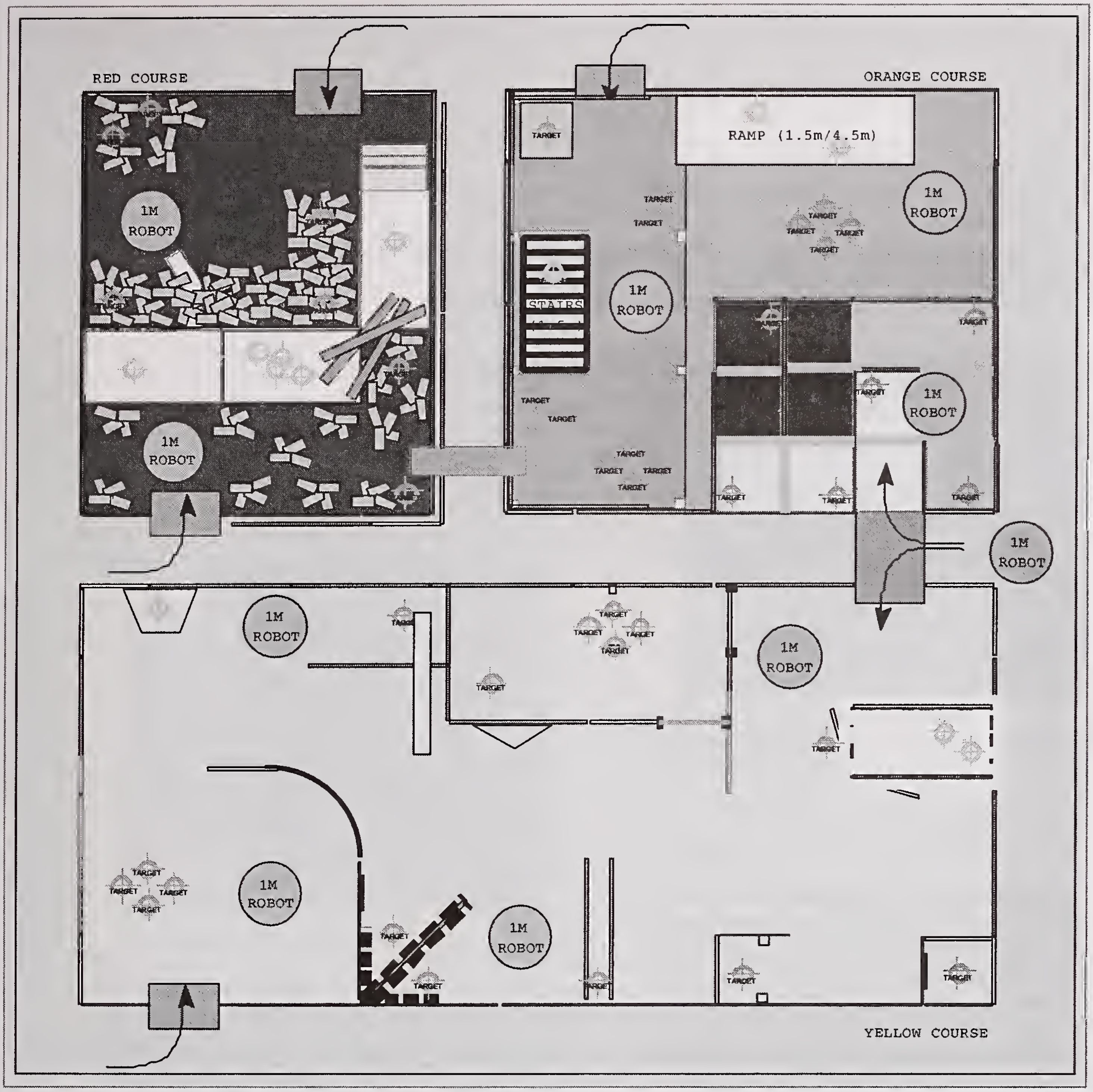


Figure 3: Overall USAR Test Course Layout

Overall dimensions are approximately 17 by 20 meters

1M ROBOT — 1 Meter Robot drawn to show scale

⌖ — Represents a victim "signature", such as a thermal emission, clothing, or other manifestation

Assessment of the NIST Standard Test Bed for Urban Search and Rescue

Robin Murphy, Jenn Casper, Mark Micire, Jeff Hyams
Computer Science and Engineering
University of South Florida
Tampa, FL 33620
{murphy, jcasper, mmicire, hyams}@csee.usf.edu

ABSTRACT

The USF team in the 2000 AAAI Mobile Robot Competition had the most extensive experience with the NIST Standard Test Bed for USAR. Based on those experiences, the team reports on the utility of the test bed, and makes over 20 specific recommendations on both scoring competitions and on future improvements to the test bed.

1 INTRODUCTION

A team of three operators and two robots from the University of South Florida (USF) tested the NIST standard test bed for urban search and rescue (USAR) as part of the 2000 AAAI Mobile Robot Competition USAR event. The test bed consisted of three sections, each providing a different level of difficulty in order to accommodate most competitors (see Fig. 1). The easiest section, Yellow, contained mainly hallways, blinds, and openings to search through. The course could be traversed by a Nomad type robot. The intermediate Orange Section provided more challenge with the addition of a second level that was reachable by stairs or ramp. Other challenges included those found in the yellow as well as some added doors. The Red Section was intended to be the most difficult. It contained piles of rubble and dropped floorboards that represented a pancake-like structure. The Orange and Red sections clearly required hardware that was capable of traveling such spaces.

In addition to USF, three other teams entered the AAAI competition's USAR event: Kansas State, Swarthmore College, and University of Arkansas. The Kansas State team dropped out due to hardware failures on site. The Swarthmore and Arkansas teams fielded Nomad scout types of robots that operated only in the Yellow Section. The performance of each team is unclear as the judges did not record how many victims were found and how many victims were missed. At the time of publication of this paper, the awards for the event were under protest. Swarthmore had a single robot which attempted to enter a room, perform a panoramic visual scan for possible victims, mark the location on a map, and then enter another room and so on. At the conclusion of their allotted time, the robot was retrieved and the contents of the map was made available to the judges. They entered one room successfully and it is believed they identified up to two surface victims. The Arkansas team used two Nomad scout type robots; however, each robot was physically placed in a room, and the team was allowed to repeatedly move and reset the robots as needed. The Arkansas team found at least one victim, and communicated this by repeatedly ramming the mannequin.

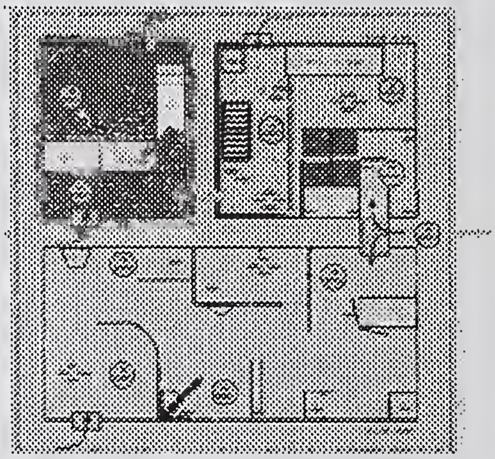


Figure 1: Overview of the NIST USAR arena.

The USF team used two outdoor robots: 1) a RWI ATRV with sonar, video, and a miniature uncooled FLIR and 2) a customized RWI Urban with a black and white camera, color camera, and sonars. This was intended to be a marsupial pair, but the transport mechanism for the team was still under construction at the time of the competition. The USF team used a *mixed-initiative* or *adjustable autonomy* approach: each platform was teleoperated for purposes of navigation but ran a series of software vision agents for autonomous victim detection: motion, skin color, distinctive color, and thermal region. The user interface displayed the extraction results from each applicable agent and highlighted in color whenever the agent found a candidate. A fifth software agent ran on the ATRV which fused the output of the four agents, compensating for the physical separation between the video and FLIR cameras. It beeped the operator when it had sufficient confidence in the presence of a victim, but the beeping had to be turned off due to a high number of false positives generated by the audience. The ATRV found an average of 3.75 victims per each of the four runs recorded, while the Urban found an average of 4.67 victims. A fifth run was not recorded and no data is available.

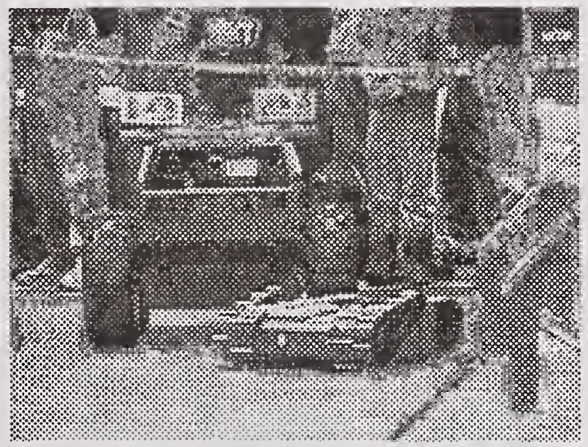


Figure 2: The USF USAR robot team, Fontana (ATRV) and Klink (Urban) (named after two women Star Trek writers).

In addition to participating in the competition (both a preliminary and a final round), the USF team hosted three complete exhibition runs as part of the AAAI Robot Exhibition Program and did numerous other partial exhibitions for the news media at the request of AAAI. The other teams did not exhibit. As such, the USF team had the most experience with the most difficult sections of the test bed and can claim to represent user expertise.

This paper discusses the NIST test bed from the USF experience, and makes recommendations on scoring, improving the test bed, and staging a more USAR-relevant event at RoboCup Rescue in 2001.

2 ASSESSMENT OF THE THREE SECTIONS

The NIST test bed is an excellent step between a research laboratory and the rigors of the field. For example, USF has a USAR test bed (Fig. 3), but it is somewhere between the Yellow and Orange sections in difficulty and cannot provide the large scale of the NIST test bed. One advantage is that the test bed sections can be made harder as needed. An important contribution that should not be overlooked is that the test bed appeared to motivate researchers we talked to: it was neither too hard nor too trivial. This is quite an accomplishment in itself.

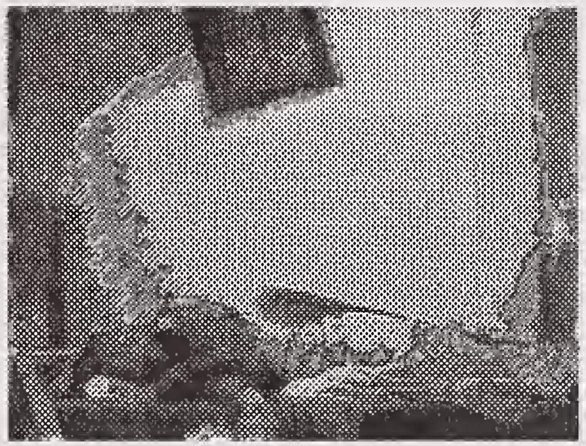


Figure 3: The USF USAR testbed, a mock-up of a destroyed bedroom.

2.1 *Yellow*

The USF team did not compete or exhibit in the Yellow Section, entering only for about 1 hour of practicing collaborative teleoperation. Our assessment was that the section was far too much of an office navigation domain- the over-turned chair in one of the rooms was the only real surprise. Only one room had a door and neither Swarthmore nor Arkansas reached it. The arena was at about the level of complexity seen in the Office Navigation Event thread of the AAAI Robot Competition in the mid-1990's.

2.2 *Orange*

The Orange Section consisted of a maze plus a second story connected by a ramp and stairs. Unlike the Yellow Section, the doorways into the Orange and Red Sections had cross-members crowning the doorway at about 4 feet high. This added some feel of confined space. The USF robots entered a very confined maze of corridors to find a surface victim. The Urban served as point man, exploring first, then guiding the ATRV if it found something requiring confirmation or IR sensing. The maze had hanging Venetian blinds in the passage way, and the Urban almost got the cord tangled in her flipper.

The Orange Section also had two forms of entry in the main search area after the robots had navigated the maze. One entry was through the X made by cross-bracing the second story. The Urban could navigate under the cross-bracing, but the ATRV could not. The second form of entry was through a door on hinges. The Urban pushed the door open for the ATRV to enter the main search space (Fig. 4). The Urban attempted to climb the stairs, but the first step was too high for the design. (A Matilda style robot also attempted to climb the stairs but could not either.) It went to the ramp and climbed to the second story.

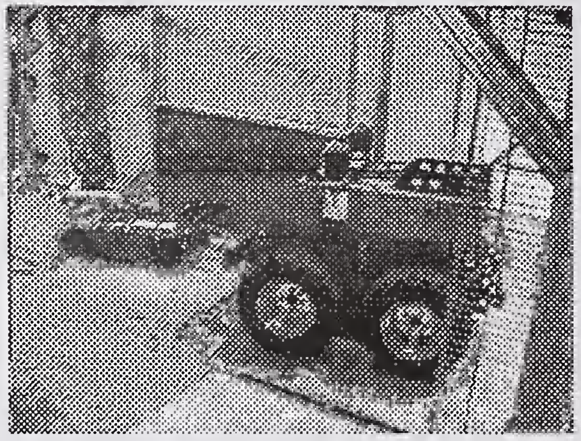


Figure 4: The Urban holds the door for the ATRV in the Orange Section.

The USF robot was able to avoid negative obstacles (a stairwell and uncovered HVAC ducting in the floor of the second story) to find victims on the second story (Fig. 5). The modified Urban actually flipped its upper camera onto the HVAC hole and peered inside the duct. This shows the utility of having multiple sensors and in different locations.

The Orange Section is also to be commended for providing some attributes of 3D or volumetric search. For example, an arm was dangling down from the second story and should have been visible from the first floor. Note that the dangling arm posed a classic challenge between navigation and mission. The mission motivates the robot or rescuer to attempt to get closer and triage the victim, while the navigational layout prevents the rescuer from approaching without significantly altering course, and even backtracking to find a path.

2.3 *Red*

The Red Section at first appeared harder (Fig. 6), however, in practice it was easier for the ATRV than the Orange Section due to more open space. The floor was made up of tarps and rocks on plywood. The ATRV and Urbans were built for such terrain. The Red Section contained two layers of pancaking, with significant rubble, chicken wire, pallets, and pipes creating navigation hazards for the Urban. Only about 30% of the area was not accessible to the larger ATRV due to the large open space.

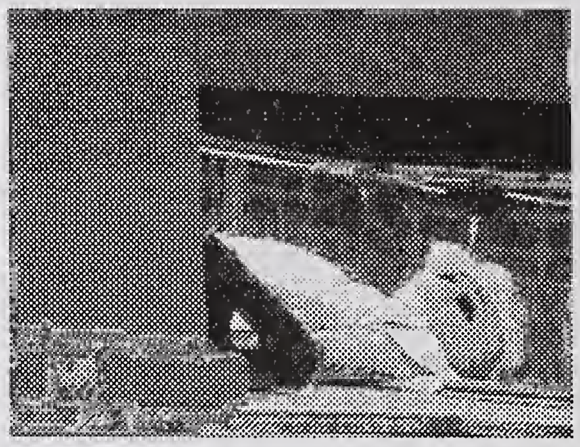


Figure 5: Close up of victim lying on the second floor of the Orange Section.

One nice attribute of the Red Section is that it lends itself to booby-traps. The pancake layers were easily modified between runs to create a secondary collapse when the Urban climbed to the top. Using current technology, the Urban operator and/or software agents could not see any signs that the structure was unstable.

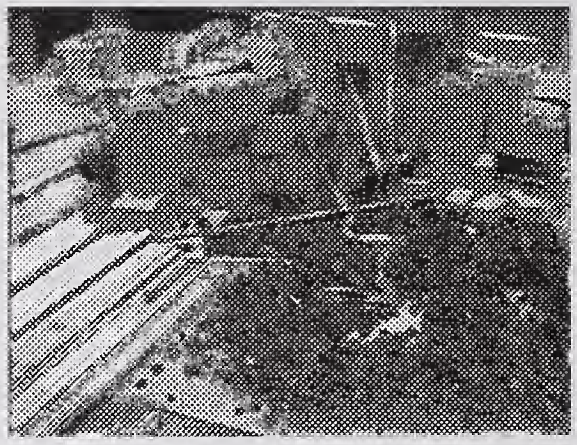


Figure 6: Overview of the Red Section.

3 RECOMMENDATIONS ON SCORING

The AAAI Competition did not use any metric scores for their USAR event, relying entirely on a panel of four judges, none of whom had any USAR experience. The AAAI Competition had published metrics prior to the competition that were to be used in scoring,[5] but did not use those metrics on site and the scoring was subjective. The published metrics appeared to be a good first start (with our reservations given below) and no reason was given why AAAI abandoned them.

1. Use quantitative scoring, at least as a basis for the com-

petition. The scores might be modified by a qualitative assessment of the AI involved, but there should be a significant numerical aspect to the scoring.

2. Distribute victims in same proportions as FEMA statistics given in FEMA publication USFA/NFA-RS1-SM1993 and award points accordingly. Detecting a surface victim and an entombed victim require much different sensing and intelligence.

Surface	50%
Lightly trapped	30%
Trapped in void spaces	15%
Entombed	5%

3. Have a mechanism for unambiguously confirming that the victims identified were identified. It was not clear to the audience when a victim had been correctly detected or when the robot had reported a false positive. Perhaps an electronic scoreboard showing the number of false positives and false negatives (missed victims) could be displayed and updated during the competition. (Swarthmore used beeping and USF flashed the headlights. The judges appeared to accept that if there was a victim in the general direction of the robot's sensors at the time of the announced detection that a victim had been found. In the case of USF, only one judge took time during the competition look at the technical rescue display workstation, which provided both the sensor data and the fused display, to confirm what the robot was seeing.)

4. Points for the detection of a victim should also depend on the time at which the technical rescue crew is informed of the discovery and the accuracy of the location, either in terms of absolute location or a navigable path for workers to reach the victim. Robots which overcome inevitable communications problems by creating a relay of "comms-bots" or returning to locations where broadcasting worked are to be rewarded. (The Swarthmore robot beeped when it thought it found a victim, but in terms of truly communicating that information to rescue workers, it stored the location of all suspected victims until the competition was ended. In practice, if the robot had been damaged, the data would have been lost. Also, the map was not compared quantitatively to the ground truth.)

5. Contact with the victims should be prohibited unless the robot is carrying a biometric device that requires contact with the victim. In that case, the robot should penalized or eliminated from competition if contact is too hard or otherwise uncontrolled. (The Arkansas robots repeatedly struck the surface victim it had detected.)

6. Fewer points should be awarded for finding a disconnected body part (and identifying it as such) than for finding a survivor.

7. Require the robots to exit from the same entry void that they used for entry. This is a strict requirement for human rescue workers in the US, intended to facilitate accounting for all resources. (The AAAI Competition permitted exiting from anywhere on the grounds that the robot may need to find a clear spot to communicate its results.)

8. Have all competitors start in the same place in the warm zone, and do not permit them to be carried by human operators inside the hot zone. The exception is if the robot has to be carried and inserted in an above grade void from the outside. (Swarthmore and Arkansas manually placed their robots in the yellow section, with Arkansas actually placing their robots within specific rooms in the yellow section.)

9. Do not permit human operators to enter the hot zone and reset or move robots during the competition. (Arkansas team members repeatedly entered the hot zone to reboot errant robots and to physically move robots to new rooms to explore.)

10. Have multiple runs, perhaps a best of three rounds approach used by AUVSI. (NIST "booby-trapped" the Red Section after the AAAI Preliminary Round, making it extremely easy to create a secondary collapse. This was done to illustrate the dangers and difficulties of USAR. However, if the AAAI rules had been followed, this would have resulted in a significant deduction of points from the USF team, and quite a different score between runs. The difficulty of the courses should be fixed for the competition events, and changed perhaps only for any exhibitions.)

It should be clear from the above recommendations that a quantitative scoring system which truly provides a "level playing field" is going to be hard to construct. Unlike RoboCup, where the domain is a game with accepted rules and scoring mechanisms, USAR is more open. In order to facilitate the relevance of the competition to the USAR community, we recommend that scoring mechanisms be derived in conjunction with USAR professionals outside of the robotics community and with roboticists who are trained in USAR. We propose that a rules committee for RoboCup Rescue physical agent be established and include at least one representative from NIST, NIUSR, and one member of the research community who had worked and published in USAR.

4 RECOMMENDATIONS FOR IMPROVING THE NIST TESTBED

The NIST testbed was intended to be an intermediate step between a research laboratory and a real collapsed building. The three sections appeared to be partitioned based on navigability, rather than as representative cases of severity of building collapses or perceptual challenges. For example, the basic motivation for the Yellow versus the Orange and Red Sections appeared to be to engage researchers with traditional indoor robot platforms (e.g., Nomads, RWI B series, Pioneers, and so on). An alternative strategy might be to consider each section more realistically, where the Yellow Section would be a structurally unsound, but largely navigable, apartment building, the Orange Section might be an office building in mixed mode collapse such as many of the buildings in the 1995 Hanshin-Awaji earthquake, and the Red Section might be a pancake collapse such as seen in the front of the Murrow building at the Oklahoma City bombing. This approach would permit traditional indoor robot platforms to navigate, but require advances in detection of unfriendly terrain such as throw rugs or carpet, doors, etc.

4.1 *For All Sections*

In addition to the suggestions made above, we offer some possible improvements to the test bed:

1. Create void spaces in each section more typical of USAR (Fig. 7). In particular, there were no lean-to and V void spaces in any of the 3 sections. The red section did have some light pancaking. Victims in even the Yellow Section should be placed behind furniture and occluded by fallen furniture or even sheet-rock or portions of the ceiling.

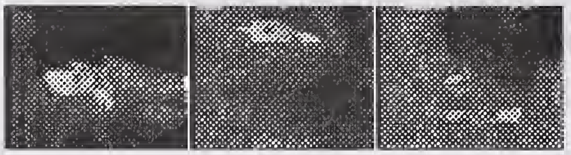


Figure 7: Infrared images of a lightly trapped, void trapped, and entombed victim.

2. Put tarps and high powered lights ("beams of sunlight") over portions of all courses to create significant changes in lighting conditions, most especially darkness. As it stands now, the testbed is a poor test of the utility of infra-red.

3. Entries were all doors at grade. Many voids are actually above grade, irregular, and have been knocked in the wall, even in buildings that have not collapsed. Each section should have one or more above grade entry voids from the "outside". This will support the testing of concepts for automating the reconnaissance and deciding how to deploy resources, as per the rescue and recovery of lightly trapped victims, use of reconnaissance results to locate lightly trapped victims, and searching void spaces after hazard removal phases of a structural collapse rescue.[4]

4. Each section should contain more human effects. For example, the Yellow and Orange Sections should have throw rugs on the floors, fallen debris such as magazines, books, bills, toys, etc. Otherwise, the Yellow Section is actually easier than the Office Navigation thread in the AAAI competitions during the mid-1990's.

5. Each section should contain real doors with door knobs or at least the commercial code handles for disabled access. The doors in the Yellow and Orange section were both easily opened panels. (USF was able to easily identify the swinging door in the Orange Section and use the Urban to open the door for the ATRV to pass through. None of the other teams got to the room with the door in the Yellow Section). All rooms in any section should have doors and some of those doors should be off their hinges or locked. This will test the advances in object recognition, reasoning, and manipulation.

6. If possible, victims should produce a more realistic heat profile than a heating pad. This is needed for detection and to test advances in assessment of the context of the victim (how much they are covered, etc.).

4.2 *For the Orange and Red Sections*

1. Cover everything with dust to simulate the cinder block and sheet-rock dust that commonly covers everything in a building collapse. Victims who are alive often move enough to inadvertently shake off some of this dust, making color detection a very important component of victim detection. (USF used a "distinctive color detector" as one of their four vision agents. The distinctive color agent looked for regions of color that were different than the average value. This appeared to work during the competition for the Red Section, which was less colorful (no wallpaper, etc.), but there wasn't enough data to draw any statistical conclusions.)

2. Make the surfaces uneven. All the surfaces were level

in their major axis; even the ramp in the Orange Section was flat, not canted to one side.

3. Use real cinder blocks. The USF Urban was able to move the faux cinder blocks on the ramp in the Orange Section rather than navigate around (Fig. 8).

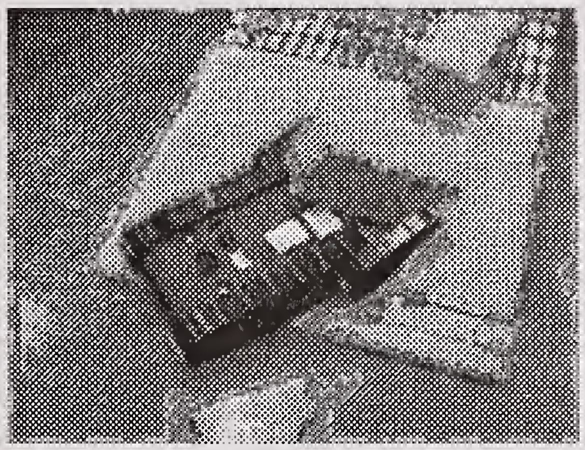


Figure 8: The Urban has pushed the cinder block around rather than traversed over it.

4. Make a "box maze" for entry to introduce more confined space. Rescue workers who are certified for confined space rescue use a series of plywood boxes which can be connected together to form long, dark, confined mazes. The mazes are easily reconfigured. A similar box maze could be constructed from the lightweight paneling material.

5. The terrain of both sections was still fairly easy compared to the field, and dry. Perhaps as robot platforms evolve, the courses should contain water.

5 OTHER SUGGESTIONS

The testbed is primarily intended to be a standard course for experimentation. The AAAI Competition did not especially further experimentation, as that the competition judges collected no metric data. However, the AAAI Competition performed a valuable service by illustrating the potential conflict between science and exhibitions. The public viewing interfered with testing and validating aspects of AI in two different ways. Public viewing may also lead to a tendency towards "cuteness" at the expense of showing direct relevance to the USAR community.

5.1 *Viewing versus Validation*

The conflict between spectator viewing and validation is best seen by the following example. One of the USF vision agents identified large regions of heat using a FLIR, then fused that data with regions of motion, skin color, and distinctive color extracted by software agents operating on video data. If there was a sufficiently strong correlation, the operator interface began beeping to draw the operator's attention to the possibility of a survivor. (The RWI supplied user interface for the Urban requires almost full attention just to navigate, detection assistance is a practical necessity.)

Unfortunately, the test bed has Plexiglas panels to facilitate judge and spectator viewing. AAAI permitted spectators to ring the sections during the competition. Between the low height of walls and the Plexiglas, these spectators were visible and produced color, motion, and IR signatures even when the USF robots were facing interior walls due to views of exterior walls in other sections. As a result, USF had to turn off automatic victim notification through audio and rely strictly on color highlighting in the display windows.

A long-term solution is to insert cameras into the testbed to record, map, and time robot activity as well as broadcast the event to a remote audience. The competition chair stated that the audience should be allowed viewing access on the grounds that rescue workers would be visible in a real site. We note that at a "real site", access to the hot zone is strictly controlled and very few, certified technical rescue workers are permitted in the hot and warm zones. The rest must wait in the cold zone at least 250 feet from the hot zone.[4] Also, at a real site, walls would have blocked views of people versus the half height panels.

Second, in order to record and broadcast the event, photographers and cameramen were permitted in the ring during the exhibitions and competition. During the exhibition, a cameraman repeatedly refused to move out of the robots' way. When the robot continued on, it almost collided with the video recorder.

Therefore, we recommend:

1. At least the Red Section should be fitted with walls and ceilings to block the view of non-testbed elements and the audience.

2. The test bed sections should be fitted with cameras and no one should be permitted in the test bed during timed events. If a robot dies (such as the USF Urban due to a faulty power supply or the Arkansas robots due to software failures), the robot should remain there until the session is complete.

5.2 *Relevance to the USAR Community*

In our opinion, the AAAI Competition missed several opportunities to show a clear relevance of the NIST test bed and robots to the USAR community. As discussed earlier, USAR professionals should be involved in setting the rules as well providing realistic scenarios. In general, any further competition venues, such as RoboCup Rescue, should actively discourage anything that might be construed as trivializing the domain. For example, Swarthmore costumed their robot as a Florence Nightingale style nurse, which rescue workers were likely to find offensive. Likewise, a handwritten "save me" sign was placed next to a surface victim.

The test bed may also miss relevance to the USAR field if it focuses only on benchmarking fully autonomous systems rather than on more practicable mixed-initiative (adjustable autonomy) systems. The Urban type of robot in a hardened form capable of operating in collapsed structures must be controlled off-board: they do not have sufficient on-board disk space to store vision and control routines. Therefore, the test bed should measure communications bandwidth, rate, and content in order to categorize the extent of a system's dependency on communications. Also, the test bed should include localized communications disrupters to simulate the effect of building rubble on communications systems.

6 CONCLUSIONS

Based on our five complete runs in the NIST test bed at AAAI and numerous informal publicity demonstrations, the USF team has had the most time running robots in the test bed. We conclude that the NIST test bed is an excellent halfway point between the laboratory and the real world. The test bed can be evolved to increasingly difficult situations. The initial design appears to have focused on providing navigational challenges, and it is hoped that future versions will add perceptual challenges.

Our recommendations fall into four categories. First, scoring or validation will be a critical aspect of the test bed. The AAAI competition did not implement a quantitative scoring system and thus provides no feedback on what are reasonable metrics. We recommend many metrics, but our guiding suggestion is to get knowledgeable representatives from the USAR community involved in setting up scenarios and metrics. In particular, we note that the victims should be distributed in accordance to FEMA statistics for surface, lightly trapped, void trapped, and entombed victims, and then points awarded accordingly. One major issue that arose from the USF team trying to reconstruct its rate of victim detection was that there needs to be an unambiguous method for signaling when a victim has been detected. Another aspect of scoring is to complement the proposed AAAI "black box" (external performance) metrics with a rigorous "white box"(software design and implementation) evaluation. Second, the test bed should be made more representative of collapsed buildings. We believe this can be done without sacrificing the motivation for the different sections. For example, all sections need to have void spaces representative of the three types discussed in the FEMA literature (lean-to, V, and pancake). The Yellow Section can still have a level, smooth ground plane but the perceptual challenges can be more realistic. Third, the test bed should resolve the inherent conflict between spectator viewing and validation. We believe this can be done by inserting cameras into the test sections as well as adding tarps and walls. Finally, we strongly urge the mobile robotics community to concentrate on making the NIST test bed and any competition venue which uses the test bed to be relevant to the USAR community. The community should resist the tendency to "be cute" and instead use the test bed as a means of rating mixed-initiative or adjustable autonomy systems that can be transferred to the field in the near future as well as the utility of fully autonomous systems.

ACKNOWLEDGMENTS

Portions of the work reported in this paper were conducted as part of work under the DARPA BAA 9835 program and through a donation from SAIC. The authors would like to thank AAAI, the USF Department of Computer Science and Engineering, and the USF Engineering College for travel support, and Mark Powell and Brian Minten for their help in programming and maintaining the robots.

References

[1] *Rescue Systems 1.* National Fire Academy, 1993.

[2] *Technical Rescue Program Development Manual.* United States Fire Administration, 1996.

[3] *Standard on Operations and Training for Technical Rescue Incidents.* National Fire Protection Association, 1999.

[4] Casper, J., Micire, M., Murphy, R.R. "Issues in Intelligent Robots for Search and Rescue," in SPIE Ground Vehicle Technology II, 2000.

[5] http://www.cs.swarthmore.edu/ meeden/aaai00/ contest.html#usar

PERFORMANCE METRICS IN THE AAAI MOBILE ROBOT COMPETITIONS

Alan C. Schultz
Naval Research Laboratory
Code 5515
Washington DC 20375
schultz@aic.nrl.navy.mil

Abstract

In this paper, performance metrics used in the AAAI Mobile Robot Competition and Exhibition over the nine years of the contest are compared. Performance metrics have tended from more explicit quantitative measures to more qualitative measures. The author believes that this trend is the result of more complex tasks where more aspects need to be measured. The paper will end by claiming that competitions that are to measure intelligence in robots should include tasks that require adaptation and learning, which the author believe are the hallmarks of intelligence.

Keywords: mobile robot contests, competitions, metrics, multi-agent robotics, learning and adaptation, autonomous robots.

1. Introduction

Although there are several annual mobile robot competitions, the American Association for Artificial Intelligence's (AAAI) Mobile Robot Competition and Exhibition has distinguished itself by attempting to reward those contestants that show the greatest amount of "intelligence" in solving a given task [1-6].

Since this event is organized as a competition, metrics are required for measuring performance in a task that also try to measure the degree of intelligence the robot has exhibited.

Because the contest is organized under the sponsorship of the AAAI, a goal of the competition is to foster research and education in artificial intelligence. As such, tasks selected for the competition were picked because they required some level of "intelligent" behavior or knowledge representation.

2. Early Years: Quantitative Metrics

In the first two years of the competition, less explicit quantitative metrics were used. However, many teams complained that the rules were not explicit enough leading to ambiguities in scoring and in problems interpreting the rules. Starting in the third and subsequent years of the competitions, more explicit and published quantitative measures of performance in the task have been used. It was assumed that completion of the task itself was indicative of intelligence. Points would be awarded to various activities (subtasks) and for abilities and competencies achieved by the robot. The final score would be a summation on the individual points. In some events, points could be removed for exhibiting some undesired behavior. Depending on the task, time would be factored into the score so that achieving the goal faster would generate more points.

The critical point is that every task and competence had points that were on a comparable absolute scale. A robot missing some skill could still win the competition. The point system would be published before the event so that teams knew exactly what score they could obtain (given that their robot performed as designed). This also allowed teams to make design decisions about what to implement on their robot.

One problem with the explicit quantitative scoring is trying to properly assign the proper score to the various competencies. As observed by Reid Simmons in the third competition [6], the virtual manipulation penalty [for not using real manipulation] "was much too small, providing a big disincentive for actually trying to grasp objects."

Another problem with using an explicit metric has been "gaming," where teams tailor their approaches to maximizing the metric. In some cases these high scoring entries violated the spirit of the particular competition. It was possible to exploit the metric in ways that gave less "intelligent" robots advantages in scoring.

Here is an illustrative example. Consider a "smart" robot that successfully exhibits all of the competencies; that is, it performs all of the aspects of the task itself, autonomously. The only problem is that this robot is slow, because of all of the processing. Now consider a not-quite-as-smart robot. Much less competent than the smart robot, it explicitly skips parts of the contest, gets help from the human, and consequently gets less competency points. But its so much faster that the overall total number of points is higher. In essence, speed wins even though part of why it was faster was because it skipped the slower, harder parts of the task.

To prevent these problems, it is necessary to design point systems where competencies define strict boundaries where lower level competencies cannot outscore higher-level competencies. But in complex tasks, this can become difficult to achieve.

3. Recent Years: Qualitative Metrics

In recent years, as the scope of the tasks in the contests have become more complex, we have found explicit quantitative metrics more difficult to implement, while at the same time having a desire to reduce gaming.

There are two reasons why the added complexity in the tasks have lead to difficulty. First, the tasks generally have multiple, sometimes conflicting aspects, and second, some of the required competencies are difficult to measure quantitatively themselves.

Human-robot communications is one competency that has proved difficult to judge in some domains. As observed in the second competition, "...because robots must often interact with humans, we tried to emphasize communications between man and machine. With a few exceptions, this aspect of the competition is still disappointing, and it is difficult to design tasks that reward appropriate communication."

Starting in the seventh annual competition, an hors d'oeuvres serving contest required the robots to serve conference attendees at a reception. Human-robot interaction was an important judged competency of the robots behavior, as was how much of the reception area was covered, and whether the robots could perceive when they needed to refill their trays. Obviously a single explicit quantitative metric is difficult.

Interesting, the first year of this contest, they awarded two separate awards based on different metrics. In addition to judging technical performance (which included the "intelligence," they also had a popular vote where conference attendees voted for their favorite entry. Its noteworthy that the robot that won first place in technical achievement did not win the popular vote.

We have tried various approaches that in general use more qualitative measures externally, while in some cases retaining internal quantitative metrics. In general, this means publishing more qualitative metrics, and hiding any explicit quantitative measures from the teams.

This is more of an "Olympic Figure Skating" style of scoring: a series of internal metrics are used in several categories that try to capture certain qualitative competencies. Judges, who are instructed in the qualitative aspects of these competencies, then assign a score from one to ten in each aspect, based, where possible, on an internal quantitative score. The external scores are then averaged, and each team is assigned a score from one to ten. By eliminating the external, published metrics, gaming could be avoided.

However, this style of scoring is generally difficult to implement. It also requires that the judges are carefully instructed, and that the qualitative aspects of the competencies are very well described such that teams are not mislead as to the way to achieve good scores, and to reduce the ambiguities like those that were present in the first two years of the competition.

This Olympic style of scoring is not appropriate for all competitions. For example, in the RoboCup competitions, where simulated and real robot teams compete in soccer competitions, there is a clear and natural quantitative score – the number of goals each team makes against the other.

Scoring Multiple Robots

One ongoing debate is how to measure the performance of multi-agent teams. The question is whether multi-robot entries need to exhibit better than linear improvements in performance over single robot teams.

Those who believe in super-linear improvement believe that the additional robots should introduce improvements that cannot be obtained by simply adding more robots to perform in parallel. Others believe that the proper metric involves looking at the total cost of implementing the team. Here the belief is that having multiple, inexpensive robots is equal to single expensive robots. There are several excellent articles in this proceedings on metrics for multi-agent systems.

Learning and Adaptation in Future Contests

One competency that distinguishes intelligence is the ability to learn and adapt to unanticipated events and conditions.

I would like to see competition events that require learning and adaptation in order to be most successful in the task. The learning and adaptation would not need to be directly scored, per se, but the tasks should be designed so that success is easier with those capabilities.

Although earlier competitions have stated this as a desired feature, learning usually just required building maps and learning locations of items in the environment, and adaptation was usually involved changing the robot's internal representations of the environment.

In particular, events where features of the environment change which require different sensing modalities or changes in strategies would allow for real indication of a robots "intelligence." Allowing judges to introduce failures in robots capabilities would be an ultimate test of the robots capability to adapt!

Acknowledgements

I want to thank Illah Nourbakhsh for his comments on previous AAAI contests. I also wish to thank all of the people who have contributed to past contests, either as organizers, sponsors, or as contestants.

References

[1] Arkin, R.C., 1998. "The 1997 AAAI Mobile Robot Competition and Exhibition," *AI Magazine*, 19(3), 13-17.

[2] Dean, T. and R. P. Bonasso, 1993. "1992 AAAI Robot Exhibition and Competition," *AI Magazine*, 14(1), 49-57.

[3] Hinkle, D., Kortenkamp, D., and Miller, D., 1996. "The 1995 Robot Competition and Exhibition," *AI Magazine*, 17(1), 31-45.

[4] Konolige, K., 1994. "Designing the 1993 Robot Competition," *AI Magazine*, 15(1), 57-62.

[5] Kortenkamp, D., Nourbakhsh, I., and Hinkle, D., 1997. "The 1996 AAAI Mobile Robot Competition and Exhibition," *AI Magazine*, 18(1), 25-31,1997.

[6] Simmons, R., 1995. "The 1994 AAAI Robot Competition and Exhibition," *AI Magazine*, 16(2), 19-30.

Performance Evaluation of Robotic Systems: A Proposal for a Benchmark Problem

Sunil K. Agrawal*, Armando M. Ferreira**, Stephen Pledgie**

Mechanical Engineering Department - University of Delaware

Newark DE 19716 USA - agrawal@me.udel.edu

Abstract

For highly agile autonomous systems, the dynamics plays a central role in the development of planners and feedback controllers to achieve a certain desired task. Trajectory plans that do not satisfy the system dynamics and constraints have a small likelihood for implementation without placing undue demands on the controllers. Coordinated control of such systems in groups becomes even more challenging because of the potential of dynamic interaction between members of the group, distributed nature of sensing, computation, and control. Among other desirable criteria, such as low energy consumption and constraint satisfaction, a measure of performance for robotic systems is compliance with its own dynamics and those of the other co-players in the group.

In this paper, we propose a benchmark problem for controller performance evaluation of a group of mobile robots. This benchmark experiment is inspired by a platoon of autonomous vehicles with the goal to change its formation over time. The objective is to obtain these formation changes while minimizing certain meaningful cost criteria. We assume that the physical models that describe the system are subject to errors. The sensor is not perfect and the structure of the controller has been selected by a user. For such a system, we can obtain the theoretically optimum trajectory with a measure of the cost. This cost can then be compared to the actual cost during hardware implementation on an experiment set up.

$*AssociateProfessor, **GraduateStudents$

We propose the following hardware set up with four vehicles in our Mechanical Systems Laboratory at University of Delaware. We plan to make this physical facility available to other members of the research community to test the effectiveness of their algorthims and controller implementations. Within such a facility, the different parameters of the model and controller can be altered to evaluate the performance sensitivities as a result of these change in parameters.

Our implementation on this experiment setup will be based on a two degree-of-freedom controller approach: (i) development of a reference trajectory for the system consistent with dynamics and constraints; (ii) an exponentially stable controller implemented around the reference trajectory. The reference trajectory development will be based on results from nonlinear systems theory and feedback linearization to efficiently solve the problem in a higher-order space, with a large fraction of computations done off-line ([1], [2], [3]). Such a study will bring out the issues of performance degradation during an experimental task and will provide a rich test-bed for comparing the effectiveness of different paradigms of control.

References

[1] Veeraklaew, T. and Agrawal, S. K., "A New Computation Framework for Optimization of Higher-Order Dynamic Systems", to appear in *AIAA Journal of Guidance, Control, and Dynamics*, 2000.

[2] Veeraklaew, T. and Agrawal. S. K., "Designing Robots for Optimal Performance During Repetitive Motion", *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 5, Oct 1998, 771-777.

[3] Ferreira, A. M. and Agrawal, S. K., Pledgie, S., "Planning for Autonomous Vehicle Platoons Using Differential Flatness", *to appear in Proceedings of 5th International Symposium on Advanced Vehicle Control and Control*, 2000.

A MULTI-SENSOR COOPERATIVE APPROACH FOR THE MOBILE ROBOT LOCALIZATION PROBLEM

Arnaud CLERENTIN, Laurent DELAHOCHE, Eric BRASSART
CREA
Centre de Robotique, d'Electrotechnique et d'Automatique
IUT, département Informatique, Avenue des Facultés, 80000 Amiens – France

Arnaud.Clérentin@iut.u-picardie.fr , Laurent.Delahoche@u-picardie.fr

Abstract

In this paper, we present a multi-sensor cooperation paradigm between an omnidirectional vision system and a low cost panoramic range finder system using to localize a mobile robot in its environment. These two sensors, which have been used independently until now, provide some complementary data. This association enables us to build a robust sensorial model which integrates an important number of significant primitives. We can thus realize an absolute localization of the mobile robot in particular configurations, like symmetric environments, where it is not possible to determine the position with the use of only one of the two sensors. In a first part, we present our global perception system. In a second part, we describe our sensorial model building approach and our segment classification method which takes into account the belief notion concerning a sensor. Finally we present an absolute localization method which uses three matching criteria fused thanks to the combination rules of the Dempster-Shafer theory. The basic probability assignment got for each primitive matching enables to estimate the reliability of the localization. We test our global absolute localization system on several robot's elementary moves in an indoor and symmetric environment.

1 Introduction

Autonomous mobile robots cannot rely solely on dead-reckoning to determine their configuration because dead-reckoning errors are cumulative. That's why they must use exteroceptive sensors that get information from the environment in order to estimate the robot's location more accurately. This leads to a classical localization method based on the fusion of dead reckoning data and exteroceptive data. The fusion method generally used is based on the extended kalman filter (EKF). The perception systems used both with the dead reckoning can be of different natures: a goniometer [3], the SYCLOP system [4], a laser range scanner [2].

Another approach consists in using only exteroceptive data: the robot's configuration is calculated in the environment reference without using previous information. To answer to this problem, two strategies are generally used. The first consists in marking the robot's evolution world with artificial beacons [5]. The second one consists in using the intrinsic features of the environment (doors, edges, corners...)[4] [1].

Artificial beacons can be detected fast and reliably and provide accurate positioning information with minimal processing. This kind of system is generally employed for industrial applications [10]. Unfortunately, these methods lack flexibility and modularity because it is necessary to fit out the robot's evolution environment.

The other solution consists in referencing on environment characteristic elements and offers a great modularity because the robot can localize itself directly in accordance with the landmarks. This kind of localization is founded on a matching stage between a sensorial model and a theoretical map of the environment. The perception systems used in that case are often the vision systems and the range finding ones. Perez in [6] determines with a panoramic laser range finder the absolute position of its robot by using the line segments as sensorial primitives. Similarly Yagi uses an omnidirectional vision system to develop navigation and environment map building methods [1]. We can notice that the robustness of these methods is mainly linked to the matching stage. The more precise and rich information the sensorial model will give, the more robust the matching stage will be.

That is why we have worked on a localization approach based on the cooperation of two omnidirectional perception systems: the vision system SYCLOP and a low cost range finder system. The association of these two kinds of complementary information permits to generate a sensorial model with a high descriptive level. Then, the matching stage provides an unique solution and we obtain a robust absolute determination of the robot's configuration.

The first part of this paper presents the global omnidirectional perception system. The second part deals with the sensorial model building method based on the management of two types of information. We describe also our classification method of the obtained segments on two classes according to their reliability. Our absolute localization method, based on a Dempster-Shafer multicriteria fusion approach, will be presented in the last part. In the conclusion we will analyze the experimental results reached with our mobile robot SARAH.

2 The global omnidirectional perception system

To localize our mobile robot, we use an original perception system making cooperate two omnidirectional sensors: an omnidirectional vision system (SYCLOP) [4] and a low cost and fast panoramic range finder system (Figure 1). These two sensors have been developed and used independently within our laboratory [4] [9]. The rotation axis of the laser is in line with the center of the conic reflector. This geometric constraint is taken into account at the time of a previous phase of calibration.

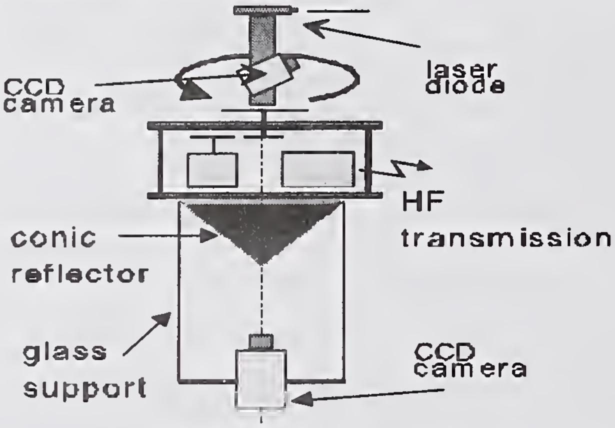


Figure 1: The global perception system

The range finder system is an active vision sensor. This methcd consists in projecting on the scene a visible light with known pattern geometry (a laser spot in our case). A camera images the illuminated scene with a given parallax. The desired 3D-information can be deduced from the position of the imaged laser point and the lateral distance between the projector and the camera (Figure 2).

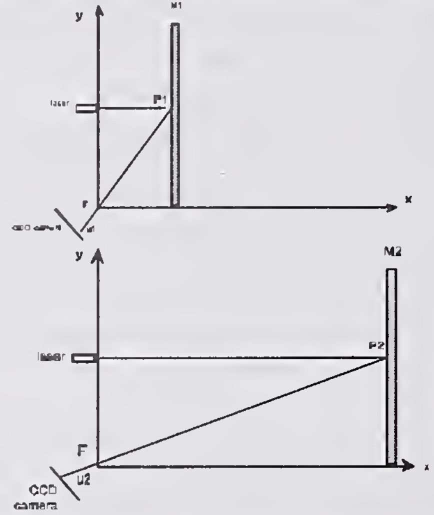


Figure 2: The geometric configuration of an active triangulation system.

The laser beam intersects the landmark M1 in the point P1 (Figure 2). This point is projected on the retinal plane through the focal point F to a point u1. A landmark M2, located at an other distance, generates a point u2. The distance of the landmark or the object Mi can be determined from the position of the point ui.

This perception system allows to obtain an omnidirectional range finding sensorial model. We manage in the sensorial model reference the cartesian distance between the laser spot and the sensor. The kind of primitives is the same that a classical range finder laser. The interest of this system is on the one hand its low cost and on the other hand its rapidity.

The prototype we built is constructed from a laser diode and a CCD camera. An infrared filter is used to extract only the light of the laser. The effective measurable distance region is designated as 0.8m-5m: this distance is thought to be a sufficient distance for a mobile robot to detect obstacles and maneuver around them.

The experimental study of this sensor is presented in [9].

The SYCLOP system, similar to the COPIS one [1], is composed of a conic mirror and a CCD camera. It allows to detect all the vertical landmarks of the environment thanks to a two dimensional projection. (Figure 3). The vertical landmarks are characterized by a radial straight line corresponding to a high contrast variation. These radial straight lines are extracted with a treatment based on the Sobel gradient. We can note that we work in fairly constraint environments, which not generate an excessive number of detected landmarks.

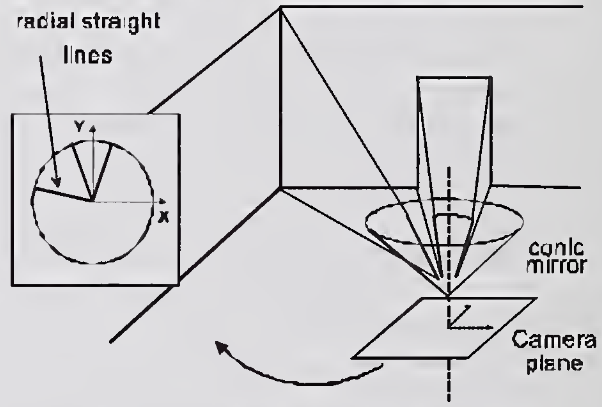


Figure 3: Principle of the omnidirectional sensor SYCLOP

This two omnidirectional sensors association permits to manage some complementary and redundant information within the same sensorial model. With the SYCLOP system we exploit, after the segmentation phase [1], the radial straight lines which characterize angles of every vertical object as, for example, doors, corners, edges, radiators. With the vision system, the information of depth cannot be gotten on an unique acquisition. For example, it is not possible to differentiate with this only sensor use the notion of opening (corridor, opening of door....) and the notion of vertical object (closed door, radiator,...) (Figure 4).

For a higher description level, it is therefore interesting to use a sensor providing some complementary information. Then we have associated to SYCLOP an inexpensive range finding sensor capable to be fast. Following a segmentation stage [9], this sensor permits us to exploit sensorial primitives that are segments (Figure 4). These segments characterize straight partitions of the environment. In this case we have the notion of depth, but it is impossible to differentiate two vertical objects placed in the same alignment: for example two closed doors placed on the same wall (Figure 4). It misses the notion of angle that will be provided by the SYCLOP system.

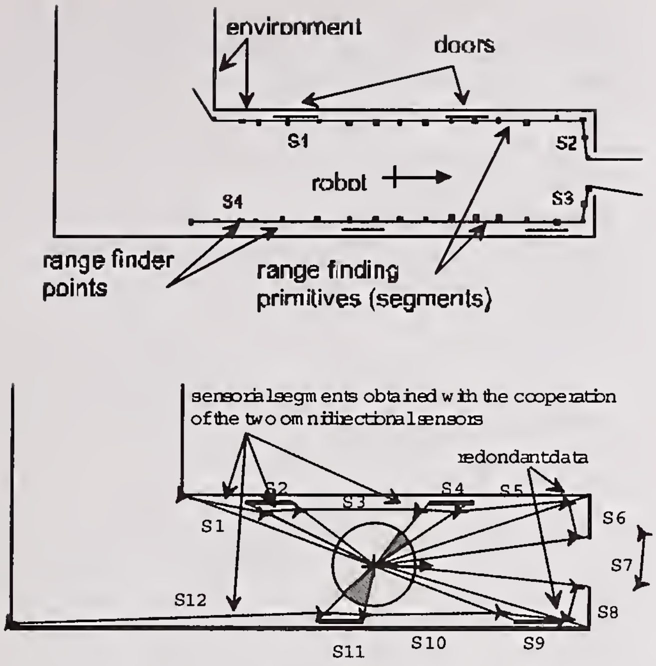

environment
doors
S1
S2
robot
S4
S3
range finder points
range finding primitives (segments)
sensorial segments obtained with the cooperation of the two omnidirectional sensors
redondant data
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12

Figure 4: Principle of the omnidirectional sensorial cooperation.

Finally this cooperative approach permits to construct a sensorial model whose descriptive level is high. This descriptive level is superior to the one obtained with each sensor individually. Moreover with an appropriated management of the redundant data (separation between two segments for the range finder and radial straight line for the SYCLOP system) we can compensate a sensorial information absence on one of this two sensors (Figure 4).

3 Sensorial Model construction

The sensorial model of the evolution world is based on the taking into account of two types of data (Figure 4): the vertical landmarks angles and the segments characterizing walls. Segments are managed with two points whose coordinates are expressed in the robot's reference. The managed primitive in the final sensorial model will be segments. These segments will be determined with two types of approaches :

- ❑ An approach based on the data complementarity: this treatment consists in cutting up segments gotten with the range finder in subsegments (Figure 5). The carving is realized with the radial straight lines of the vision system.
- ❑ An approach based on the data redundancy: the redundant aspect is characterized by the detection of a vertical landmark with the two sensors (Figure 5). In certain cases a vertical landmark is detected by the range finder with the end points of segments. We will be able to confirm the existence of a segment extremity if a radial straight line corresponds to it. In case of radial straight line absence we will keep the segmentation obtained with the range finding sensorial model.

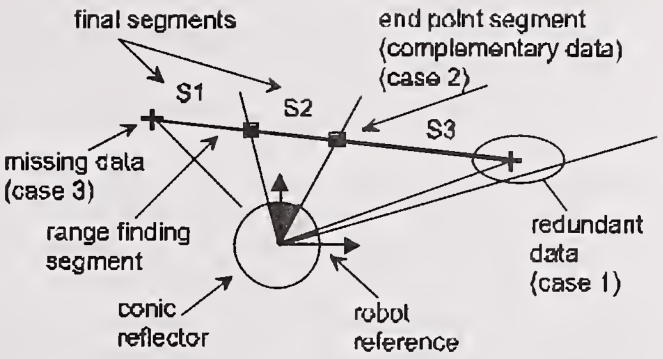


Figure 5: The Different cases of the cooperation algorithm.

We have integrated these different cases of cooperation in the sensorial model building algorithm shown on Figure 6.

The first step consists in extracting line segments from the set of points given by the sensor. We use the recursive Duda-Hart segmentation algorithm [7] [9]. To decrease the noise sensitivity of this algorithm we have added a pre-processing stage on the set of points in order to eliminate the aberrant points. Besides, in order to fit as better as possible the set of points, we apply a least square algorithm on the obtained segments.

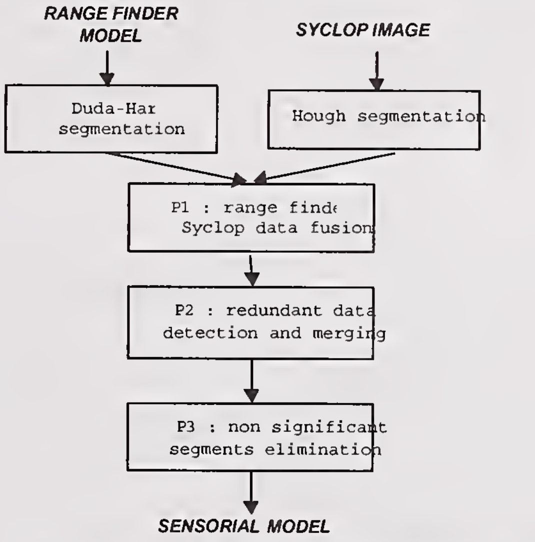

RANGE FINDER MODEL
SYCLOP IMAGE
Duda-Har segmentation
Hough segmentation
P1 : range finde Syclop data fusion
P2 : redundant data detection and merging
P3 : non significant segments elimination
SENSORIAL MODEL

Figure 6: Principle of the global sensorial model building algorithm

From the SYCLOP image, we treat the radial lines with a segmentation algorithm based on a simplified Hough transform. We fixed the threshold detection of a radial line (number of pixels composing a radial line) to an important value in order to keep the significant radial primitives.

The fusion step, described on Figure 4 and Figure 5, is based on the taking into account of three cases :

- ❑ The treatment of redundant data (case 1 of Figure 5). In this case we take as hypothesis to use the radial line systematically to determine the end point of a segment. The angle of a vertical landmark is determined more precisely with the vision system that with the range finder.
- ❑ The treatment of complementary data (case 2 of Figure 5). This treatment consists in cutting up a range finding segment into several final subsegments. This stage is based on the segment intersection determination.

- ❑ The treatment of missing data (case 3 of Figure 5). The notion of missing data is here characterized by a vertical landmark which is not detected with the vision sensor. In this case the range finding breakpoint is considered directly.

During this stage, we classify the segments and subsegments we get in two classes of reliability: a class "SURE" and a class "UNCERTAIN". In this purpose, we take into account five criterion for each segment.

The first criteria is the mean distance between the range finding points contained by the segment and this segment. If this mean distance is high, it means that the points are not very well aligned, so this segment is not very sure.

The second criteria is the number of points supported by the segment. This criteria is only discriminative when the segment contains very few points. In this case, it is not sure.

The third criteria is the segment density of points. As shown in [9], a major drawback of this kind of triangulation depth sensor is a decreasing resolution with increasing distance. So, this criteria, which is linked to the mean distance between the sensor and the set of point, is a good indicator of the segment reliability (more distant the set of points is, less the precision is). Considering the measure extent of the sensor (0.8m from 5m), the minimal and maximal density are as shown on Figure 7.

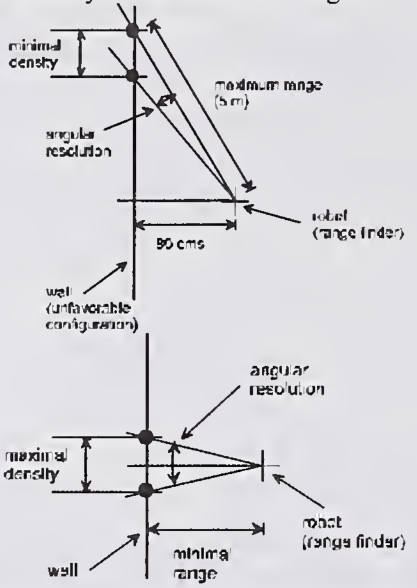


Figure 7: quantification of the density criteria

The fourth criteria analyzes if the segment has been detected by one or by the two sensors. The different cases are:

- ❑ The two extremities of a segment are detected only by the laser range finder (segment S1 in the case 1 of Figure 8). This segment has a weight of 1.
- ❑ One extremity of a segment is detected by the laser range finder and the other extremity is detected only by the conic mirror (segment S1 in the case 2 of Figure 8). This segment has a weight of 2 because, as we say before, we think that the radial straight lines are more precise and reliable.
- ❑ One extremity of a segment is detected by the two sensors and the other extremity is detected only by the laser, or the two extremities are detected only by the conic mirror (segment S1 in the case 3 of Figure 8). This segment has a weight of 3.
- ❑ One extremity of a segment is detected by the two sensors and the other extremity is detected only by SYCLOP (segment S1 in the case 4 of Figure 8). This segment has a weight of 4.
- ❑ The two extremities of a segment are detected by the two sensors (segment S1 on the case 5 of Figure 8). This segment has a weight of 5.

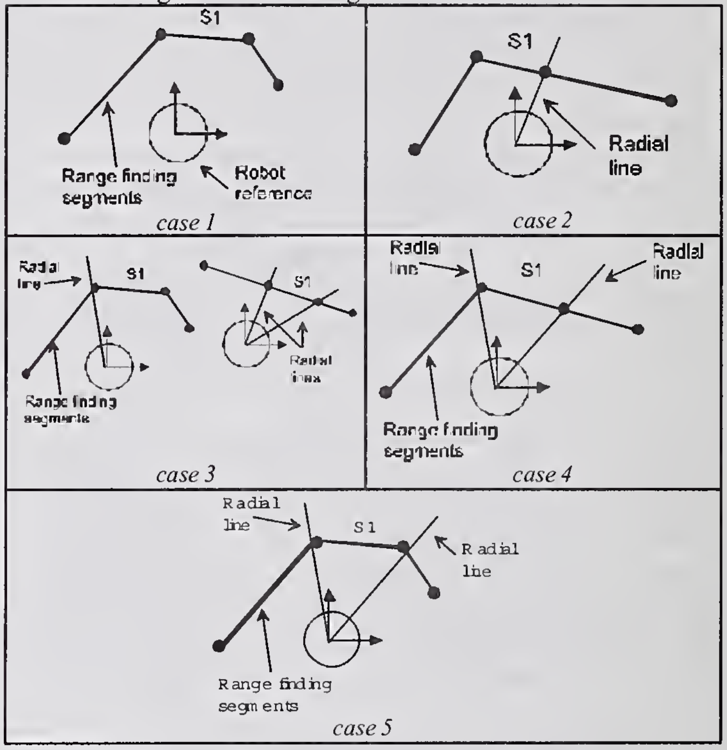


Figure 8: powdered segment, the four cases.

The last criteria concerns a gray level curves extracted from the SYCLOP image. We take into consideration five concentric gray level circles whose average is made. We obtain thus one gray level curve from 0 to 360 degrees. We apply on the portions of curve which represent a segment a least square algorithm. We obtain a straight line and we compute the mean difference of the gray level value from this line. If this mean difference is high, this means that the gray level sector is not constant. This case occurs generally when a landmark has not been detected by SYCLOP, so this segment is not sure.

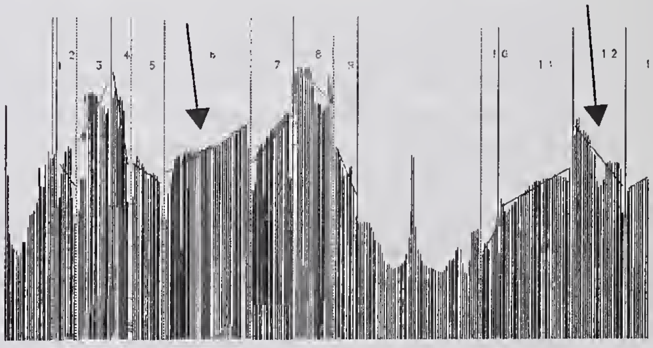


Figure 9 : the gray level curve of the experimental result shown fig. 14

The fusion of these five criteria is made thanks to the combination rules of the Dempster-Shafer theory [8][11]. We use this theory because it is an interesting formalism which enables to represent ignorance. Our frame of discernment is composed of two elements: "SURE" and "UNCERTAIN". The basic probability assignments m_1, m_2, m_3, m_4 and m_5 for this five criteria are shown in Figure 10. We can see that, for certain values, the criterion are not discriminative and Dempster-Shafer enables to represent this ignorance (for example, if the density is equal to 0.12 points/cm, this value does not permit to take a decision SURE or UNCERTAIN for this criteria).

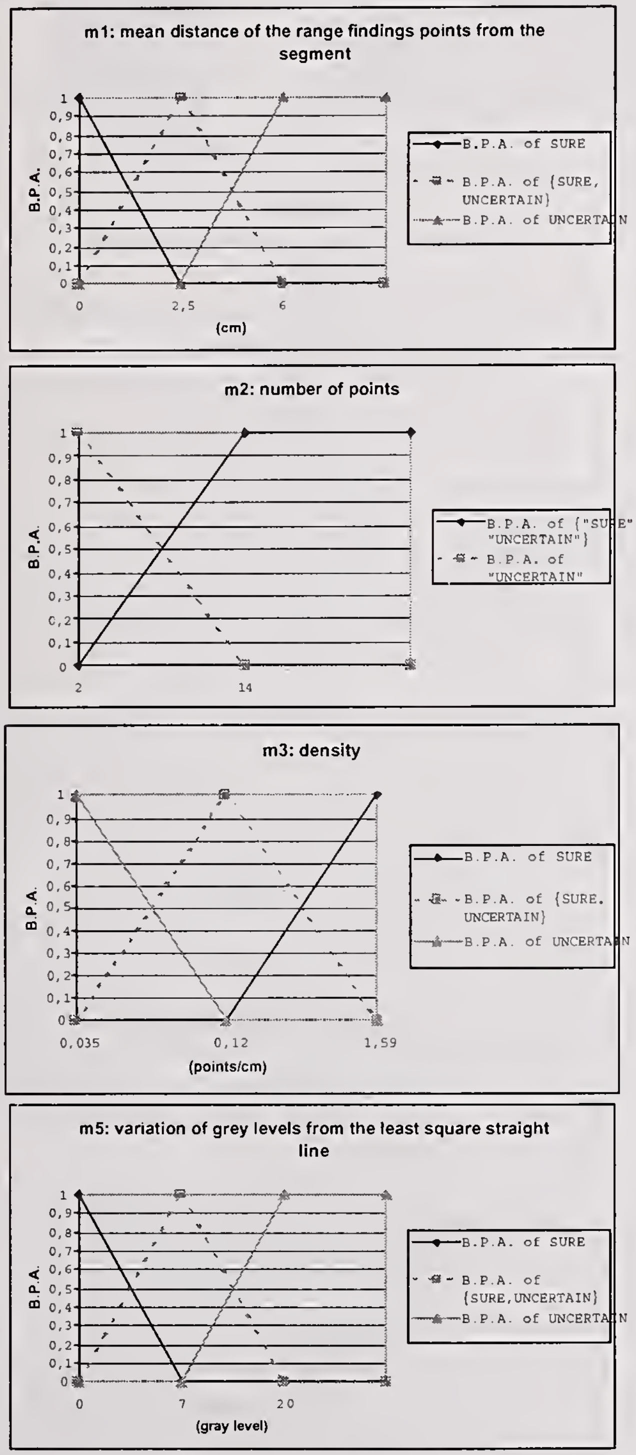


Figure 10: the B.P.As of the four classification criteria ({SURE,UNCERTAIN}=Θ)

For the fourth criteria, the B.P.A. are:
weight 1:m_4(SURE)=m_4(Θ)=0.6, m_4(UNCERTAIN)=0.4
weight 2:m_4(SURE)=0, m_4(Θ)=1, m_4(UNCERTAIN)=0
weight 3:m_4(SURE)=0.3,m_4(Θ)=0.7,m_4(UNCERTAIN)=0
weight 4:m_4(SURE)=0.6,m_4(Θ)=0.4,m_4(UNCERTAIN)=0
weight 5:m_4(SURE)=1, m_4(Θ)=m_4(UNCERTAIN)=0

We can then perform the combination calculation thanks to the Dempster-Shafer rules [8][11]. If the conflict coefficient k between the elements of the frame of discernment is superior to 0.7, it means that our criteria disagree. In this case, we decided that our segment is uncertain. If $k<0.7$, we compute the combination of belief functions for each element of the frame of discernment and we choose the class which has the maximal B.P.A.

The last stage (P3 on Figure 6) consists in eliminating the non significant segments in the final cooperative sensorial model. A non significant segment is characterized by a number of range finding points equal to 0 and a length (Cartesian distance) inferior to a predetermined threshold.

This stage permits to decrease the combinatory aspect of the matching stage and to increase the robustness.

This building algorithm enables to get a sensorial model where the number of exploitable primitives is more important than the number of primitives got by each sensor when it works individually. Besides, we obtain a certainty information of a segment by considering five criteria. This information will be used in the matching phase.

4 ABSOLUTE LOCALIZATION METHOD

The robot configuration is determined by matching the sensorial model, got by multisensor cooperation, with a theoretical map of the environment. The primitives used for this matching phase are segments. Therefore, all environment's elements like doors, walls, windows, radiators... are indexed as segments in the theoretical map.

For each segment, we have considered three correspondence tests, which are similar to these used by Crowley [7]:

- ❑ the angular difference between the two segments,
- ❑ the difference in length between the two segments,
- ❑ the distance between the centers of the two segments.

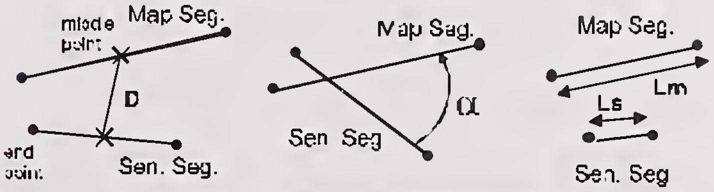


Figure 11: The three matching criteria.

The fusion of these three treatments is made thanks to the combination rules of the Dempster-Shafer theory [8]. Our frame of discernment is composed of two elements: YES and NO corresponding to those assertions : "Yes, we can match the two segments" and "No, we can not match the two segments". For each criterion, we have determined the Basic Probability Assignments (B.P.A.) m_1, m_2 , m_3 shown Figure 12.

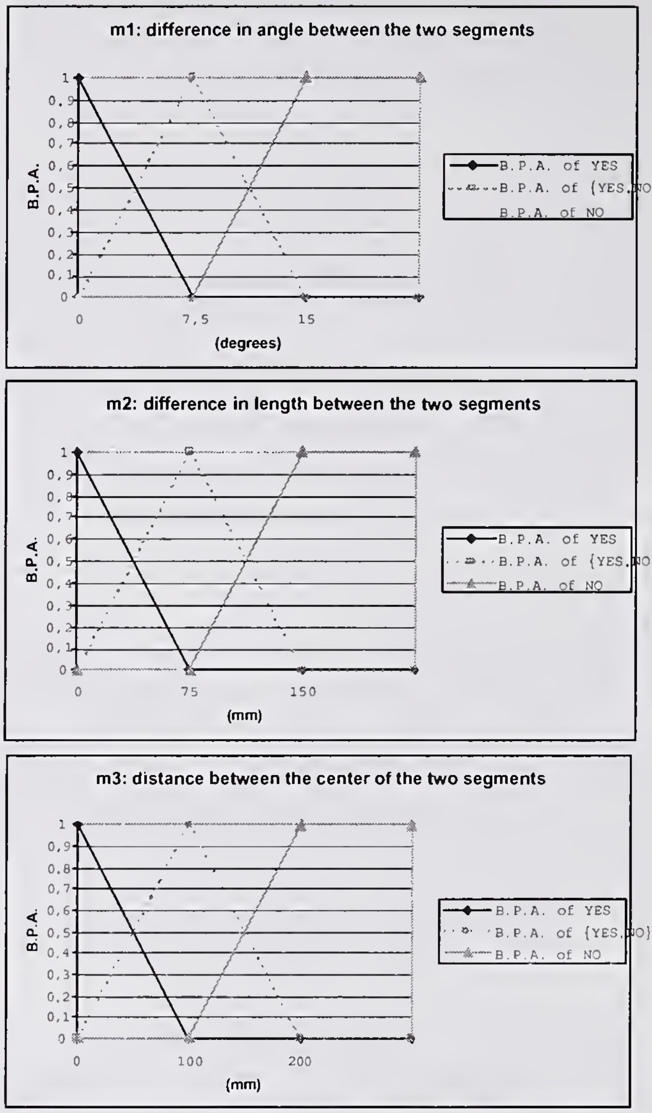


Figure 12: basic probability assignments of matching criteria ({YES,NO}=Θ)

We can then perform the combination calculation thanks to the Dempster-Shafer rules [8]. Since we have three criteria, we first fuse the two first criteria.

The conflict coefficient between these two first criteria is:

$$k_{12} = m_1(YES).m_2(NO) + m_1(NO).m_2(YES) \qquad (1)$$

If k_{12}<1, the conflict is not complete and the combination of belief functions m_{12} for these two elements of the frame of discernment is given by:

$$m_{12}\ (YES) = \frac{m_1(YES).m_2(YES) + m_1(YES).m_2(\Theta) + m_1(\Theta).m_2(YES)}{1-k_{12}}$$

$$m_{12}(NO) = \frac{m_1(NO).m_2(NO) + m_1(NO).m_2(\Theta) + m_1(\Theta).m_2(NO)}{1-k_{12}}$$

$$m_{12}(\Theta) = \frac{m_1(\Theta).m_2(\Theta)}{1-k_{12}} \qquad (2)$$

Then we fuse the last criterion. We compute the conflict coefficient k (3) between this criterion and the two criteria we have fused above:

$$k = m_{12}(YES).m_3(NO) + m_{12}(NO).m_2(YES) \qquad (3)$$

If k>0.7, we think that the conflict is too high. So we decide to take a prudent decision: we don't match the two segments. If k<0.7, we compute the combination of belief functions for each focal element:

$$m(YES) = \frac{m_{12}(YES).m_3(YES) + m_{12}(YES).m_3(\Theta) + m_{12}(\Theta).m_3(YES)}{1-k}$$

$$m(NO) = \frac{m_{12}(NO).m_3(NO) + m_{12}(NO).m_3(\Theta) + m_{12}(\Theta).m_3(NO)}{1-k}$$

$$m(\Theta) = \frac{m_{12}(\Theta).m_3(\beta)}{1-k} = \frac{m_1(\Theta).m_2(\Theta).m_3(\Theta)}{1-k} \qquad (4)$$

The segments are matched if B.P.A. for the *YES m(YES)* is superior to the B.P.A. for the *NO m(NO)*.

The first stage of this localization algorithm consists in determining a list of sensorial segments *Ls* which have a strong probability of existence. This segments are the "SURE" segments obtained during the fusion stage.

We consider that the length of these segments has been determined with a good accuracy. So, our starting correspondence test is the length of a segment.

In the second stage, we consider a segment Ls_k from the list *Ls* and we search the theoretical map segments which length is similar to the Ls_k segment length. Each found theoretical segment is superposed on the sensorial segment Ls_k and we apply the third step in order to test the correspondence of the other sensorial segments.

The third step consists in applying the three criteria describe above on all the segments on the list *Ls* except the segment Ls_k. A segment is matched if the B.P.A. for the *YES* is superior to the B.P.A. for the *NO*. To choice the optimal matching solution we calculate a *V* criteria. For each matched segment pair, we increment this *V* coefficient which characterizes the robustness of the global matching. *V* is managed with the following algorithm:

```
Given:
-    B the B.P.A. for the YES of the matched
     segment pair
-    W a weight linked to the segment's class
     (SURE segment: w=3, UNCERTAIN segment:
     w=1).
FOR each global matching
    V=0
    FOR each segment matched
        V = V + (B*W)
    END
END
```

So we can see that *V* is an interesting and discriminative indicator of the global matching relevance since *V* takes into account the class of each matched segment ("SURE", "UNCERTAIN") and the quality of each matched pair (through the B.P.A. for the *YES*).

These three steps are then repeated for all the *Ls* list segments. The final solution is the one which permits the maximal *V*.

5 Experimental results

To test the robustness of our localization algorithm, we have performed it on several sensorial acquisitions made in an indoor environment (Figure 13). The two omnidirectional acquisitions are made when the robot is stopped. The omnidirectional acquisitions and the localization algorithm are computed in a Pentium PC located on our mobile robot. A Matrox Meteor II video card is used to acquire the omnidirectionnal image and the laser acquisition. Our experimental perception system is shown on Figure 13.

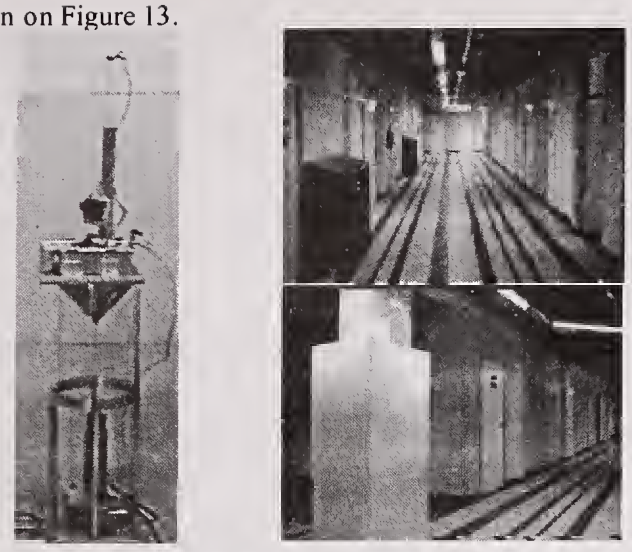
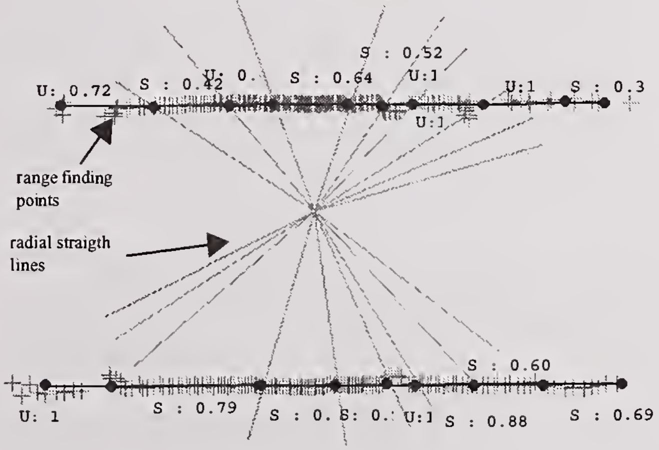


Figure 13: Our global omnidirectional perception system and the experimental indoor environment.

In order to show the interest of our cooperative approach, we have tested our localization method on symmetric environments (the first picture on Figure 13). The use of one sensor individually instead of the two sensors emphasizes the robustness problem: a strong failure rate has been observed for the matching phase when we use only one sensor [9].

The first environment is a long corridor (length: 50 meters). Figure 14 shows a sensorial model got with our cooperative approach. The robot is located in the middle of the corridor (Figure 13). We can see on Figure 14 the final decomposition on an set of segments which represent doors and parts of wall. We show on this figure the radial straight lines obtained with the omnidirectional conic mirror. We must note that, for this environment, the depth sensor would not have been able to localize the robot: two parallel identical segments would have been detected. The SYCLOP system used alone would have posed the problem of environment symmetry. We can also remark that uncertain segments are the segments which are far from the robot (not well aligned) or which correspond to the pillars of the corridor (not detected during the Duda-Hart segmentation stage). The robot final position successfully obtained shows the robustness of our method and its accuracy. We have indeed a position error of 8cm and an orientation error of 3 degree.



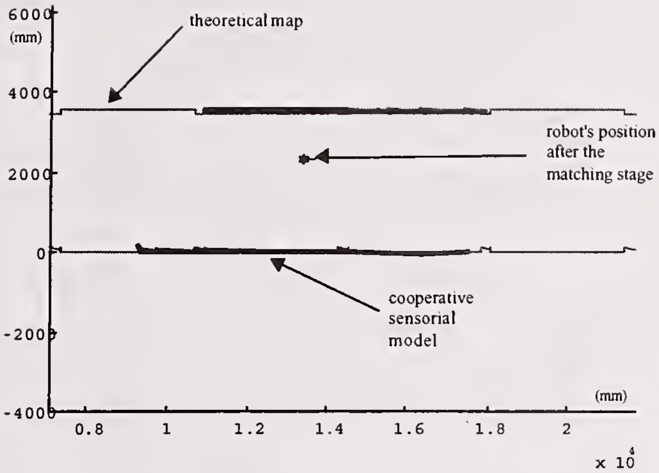


Figure 14: the cooperative sensorial model with the segments classification and the BPA(U=UNCERTAIN, S=SURE) (first figure) and the final position determination corresponding to the optimal matching

We show on Figure 15 results obtained in an other symmetric environment: a laboratory square hall.

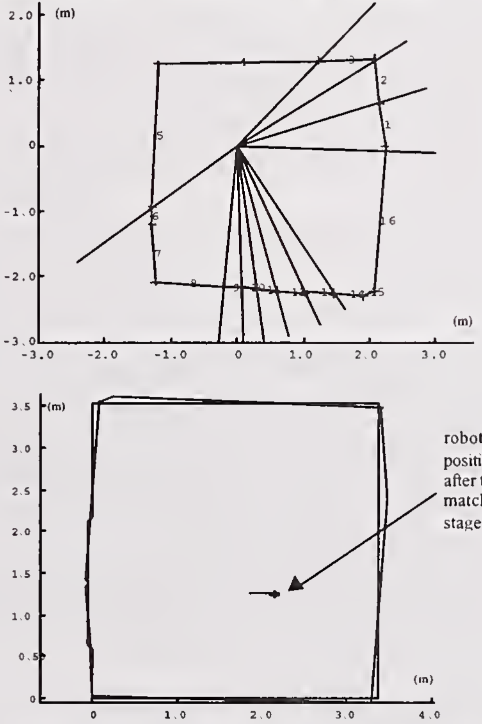


Figure 15: cooperative sensorial model and final position determination in a hall environment.

The same remarks can be done: the use of the two sensors provides enough sensorial information to enable the matching algorithm to converge to a coherent solution.

The third environment is the end of the corridor shown Figure 13. This environment constitutes a favorable experimental configuration: it is not symmetric and it has an important number of exploitable landmarks (figure 10). We can note here on several robot's configuration determination that our matching selection criteria is highly discriminative: the good configuration has been computed on all the acquisitions.

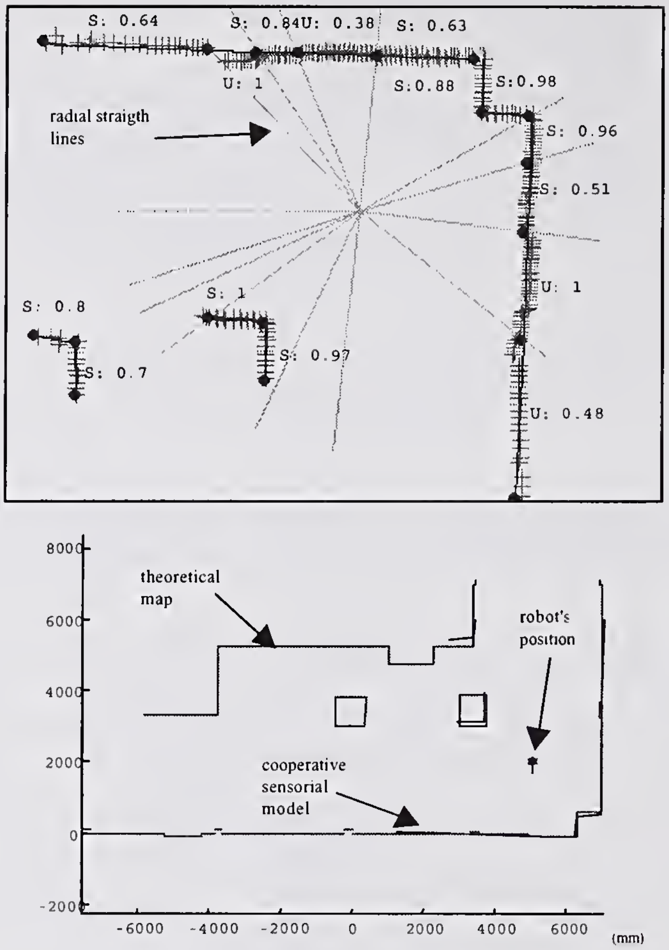


Figure 16: cooperative sensorial model (first figure) and final position determination.

Finally on a complete path makes in the corridor by our robot mobile SARAH, we could note on 40 acquisitions that, on the one hand, all the absolute configurations have been determined correctly, and, on the other hand, the mean error was equal to 11 cm in position and 3 degree in orientation.

In spite of an important combinatory aspect, our cooperative localization method proves to be robust and particularly accurate.

6 Conclusion

We have presented in this study an absolute localization approach based on the cooperation between two omnidirectional sensors: an omnidirectionnal vision sensor and a range finding sensor. This association allows to treat two types of complementary data. Then we obtain a highly descriptive sensorial model which integrates an important number of primitives and enables to increase the robustness of the matching stage. We classify also every sensed segment in two reliability classes according to five criteria fused thanks to the Dempster-Shafer rules. The absolute localization paradigm based on this matching stage takes into account several criteria which are merged with the Dempster Shafer rules. The choice of the optimal matching is based on a highly discriminative criteria which associates the segment reliability classes and a B.P.A. linked to the matching stage. We have tested our cooperative absolute localization algorithm on several particular environment like for example symmetrical environment. On the one hand, we can note on these experimental results that the robot's configuration determination is realized in a unique way and on the other hand the absolute robot's configuration is calculated with a relatively weak systematic error.

References

[1] Y. Yagi, Y. Nishizawa, M. Yachida, "Map-based navigation for a mobile robot with omnidirectional image sensor COPIS", IEEE Trans. on Robotics and Automation Vol. 11, pp. 634-648, October 1995.

[2] J.Gomes-Mota, M.I. Ribeiro, "A multi-layer robot localisation solution using a laser scanner on reconstructed 3D models", Proc. on the 6th Int. Symposium on Intelligent Robotic Systems, Scotland, 1998.

[3] P. Bonnifait, G. Garcia, "Design and Experimental Validation of an Odometric and Goniometric Localization System for Outdoor Robot Vehicles.", IEEE Trans. on Rob. and Aut. Vol. 14, No 4, pp. 541-548, August 1998.

[4] C. Drocourt, L. Delahoche, C. Pegard, C. Cauchois , "Localization method based on omnidirectional stereoscopic vision and dead-reckoning", Proc. of the IEEE Int. Conf. on Int. Robots and Systems, Korea, October 1999

[5] H.R Beom, H.S. Cho, "Mobile robot localization using a single rotating sonar and two passive cylindrical beacons", Robotica, Vol. 13, pp. 243-252, 1995.

[6] J.A. Perez, J.A. Castellanos, J.M.M. Montiel, "Continuous localization: vision vs. laser", IEEE Proc. of Int. Conf. on Rob. and Aut. pp 2917-2923, Detroit, May 1999.

[7] J. Crowley, "Navigation for an intelligent mobile robot", IEEE Journal on Robotics and Automation, Vol. RA-1, n°1, pp. 31-41, March 1985.

[8] G.A. Shafer, "A mathematical theory of evidence", Princeton : university press, 1976.

[9] A. Clérentin, C. Pégard, C. Drocourt "Environment Exploration Using an Active Vision Sensor", Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems (IROS'99), Korea ,October 1999.

[10] J. Hollingum, "Caterpillar make the earth move : automatically", Industrial Robot, Vol. 18, N° 2, pp. 15-18, 1991.

[11] A. Dempster, "Upper and lower probabilities induced by a multivalued mapping", Annals of mathematical statistics 38:325-339, 1967.

Measuring Mobile Robot Performance: Approaches and Pitfalls

Gaurav S. Sukhatme

gaurav|@usc.edu

Robotics Research Laboratories
Department of Computer Science
Institute for Robotics and Intelligent Systems
University of Southern California
Los Angeles, CA 90089-0781

Abstract

We consider the problem of measuring the performance of an intelligent mobile robot system. We believe that systems are intelligent because their capabilities are more than the sum of their parts. Therefore any piecemeal efforts to measure the performance of an intelligent system are bound to fail. Further, metrics of utility are more useful to designers than something as abstract as intelligence. We describe a task-based, multiple-criteria technique that combines two benchmarks to result in a metric for navigation. A case study of two robots is presented, which were evaluated and compared using the metric.

1 Introduction

We consider the problem of measuring the performance of an intelligent mobile robot system. We believe that systems are intelligent because their capabilities are more than the sum of their parts. Therefore any piecemeal efforts to measure the performance of an intelligent system are bound to fail. Only measuring performance along a single skill axis is also clearly limiting since intelligence does not boil down to a single skill or capability but rather arises due to a complex interplay between a multitude of capabilities. We strongly advocate the measurement of task-oriented quantities which establish the utility of a system. To this end, measuring performance along *several axes* is clearly important but brings with it several challenges:

- What should the axes be ?
- How do we ensure the axes span the space we want to benchmark ?
- Does an "orthogonal" set of axes exist ?
- How should the performance measures along these axes be combined ?

In this paper we describe a task-based, multiple-criteria technique that combines two benchmarks to result in a metric for navigation. A case study of two robots is presented, which were evaluated and compared using the metric.

2 Previous Work

Due to space limitations we limit ourselves to a brief survey of evaluation techniques for mobile robots. The so-called static evaluation techniques are specifically designed for measuring stability when the robot is stationary and when it is moving in a statically stable fashion. The primary method of choice is an energy based stability measure as an evaluation function. In work by Nagy et al. [7] two modes of walker stability are characterized namely stance stability and walker stability. Both use the amount of energy needed to destabilize the walking robot as a measure of the stability of the robot. The stance stability is identical to the energy stability margin defined by Messuri et al. in [5] as the minimum work that must be done on a robot walker to tip it over an edge of a support boundary.

Early work on robot stability was due to McGhee et al. [4] who defined the support polygon as the convex hull of the projections of all contacting points on a horizontal plane. In [3] the authors define a conservative support polygon with the motivation that the walking robot should retain its stability in the event of a single leg failure. Of the above energy based measures of stability the work of Nagy et al. is the most general since it includes compliance of the mechanism and depends on the terrain that is underfoot.

In [1] the authors discuss several evaluation criteria for comparing three configurations for the design of a walking robot. Some of the evaluation criteria were foothold selection area, stride length, static stability and energy stability. The important tradeoff was stride vs. stability, based upon which the circulating configuration for Ambler was chosen.

Dynamic evaluation techniques are so named because they focus on properties related to motion. Wilcox [10] introduced a metric called the MCC (Mobility Characteristic Curve) to measure the ability of a robot to surmount obstacles. The obstacle was a cylinder of (theoretically) infinite length and diameter d which was buried to a depth $d/3$ in an inclined plane of slope s composed of loose sand. The MCC was defined as the plot with s on the horizontal axis and the diameter of the largest cylinder that the robot could surmount (in dimensionless units based upon its length) on the vertical axis. The proposed figure of merit was the area between the co-ordinate axes and the MCC. The two main achievements of this method were its independence of scale and easy reproducibility. Its chief drawback was that it used a simple obstacle geometry and did not evaluate the entire system in a mission oriented way.

Lietzau [2] proposed a set of benchmarks to assess the performance of a Mars microrover. These benchmarks were divided into five categories namely, mobility, navigation and control, science, autonomy and environmental. A set of weights was assigned to these categories based upon their importance by the system designers and mission specialists. The weighted sum of the individual benchmarks was then proposed as a figure–of–merit. Lietzau's work is a thorough description of the individual subsystem tests that are a necessary part of evaluation but does not focus on the system level evaluation that we emphasize here. Though it was never formally characterized as such, Lietzau's evaluation technique is an example of a Linear Programming approach to solve the problem of evaluation.

3 Case Study

The evaluation methodology that we propose here is for a particular robot mission - exploration of an unknown planetary surface. The area to be explored is assumed to contain rocks whose positions are not known *a priori* to the robot since it is presumed to be in unfamiliar surroundings. The robot mission is to perform scientific experimentation on rocks that are "interesting". We propose two evaluation functions in this study based on robot displacement as a function of mission time and energy consumption.

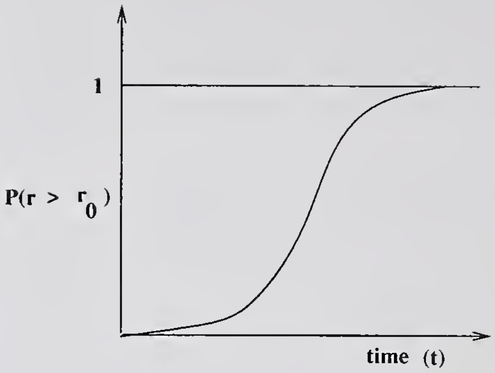


Figure 1: A Schematic of $P(r > r_0)$ vs. Time

3.1 The Cost Functions τ and η

The basic intuition behind the two cost functions proposed is to develop a nondimensional measure of the robot's ability to cover distance. The idea is to measure how "good" a particular robot design is by measuring how far the robot travels from the start location as a function of the time elapsed and the energy consumed by it. At first sight it may seem like the consumption of these two resources is extremely well correlated. This is indeed the case for straight-line travel on level ground with no obstacles. However, in the presence of obstacles it is not so - especially since the energy consumption of the system changes dramatically depending on whether it is at a standstill or in motion.

We define a trial as an autonomous traverse of the terrain by the robot in a particular instantiation of obstacle placement from start to goal. Using multiple trials we estimate the probability that the displacement $r > r_0$ for different values of the time t. A schematic of this probability as a function of time is shown in Figure 1. The main intuition is that the quicker this curve rises (close) to 1, the better the time utilization of the robot. Further, good time utilization also dictates that this curve be monotonic increasing. For the purposes of evaluation one is interested in the robot covering some displacement r_0 within some time t_0. In other words we expect some minimum performance for a limited resource (time).

The above requirement means that Robot A should be assigned a higher score than Robot B in Figure 2. This can be achieved by defining the area under the curve from $t = 0$ to $t = t_0$ as a metric. In order to compare robots of different size we measure displacement ($r = kl$) in terms of the number k of robot lengths l. We also measure time in nondimensional terms by

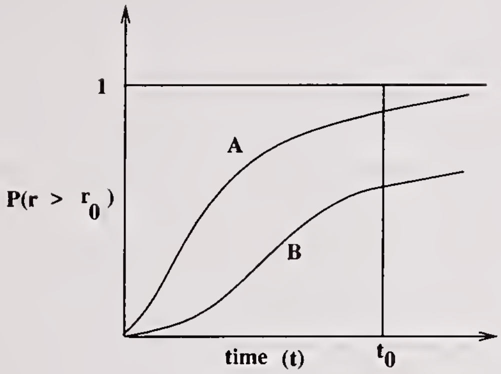


Figure 2: A Schematic Comparison of $P(r > r_0)$ vs. Time for Robots A and B

multiplying it with v/l where v is the robot velocity.

Let $\pi_{kl}(t)$ denote the probability of reaching a displacement kl as a function of time.

Definition 1 *The time figure of merit is defined as*

$$\tau = \int_0^{t_0} \pi_{kl}(t) dt \tag{1}$$

In a similar manner.we plot the probability of reaching a displacement kl as a function of the energy e consumed. Energy is converted to a nondimensional quantity by dividing it by mgl where m is the mass of the robot and g is the acceleration due to gravity.

Let $\pi_{kl}(e)$ denote the probability of reaching a displacement kl as a function of energy.

Definition 2 *The energy figure of merit is defined as*

$$\eta = \int_0^{e_0} \pi_{kl}(e) de \tag{2}$$

Note than both figures of merit are non-dimensional.

3.2 The Robots: MENO and Marscar

MENO is a 12 DOF statically stable quadruped designed and constructed for this study in the USC Robotics laboratory. Each leg is a rotary-rotary-prismatic (RRP) design. The body of the robot and the first two links of each leg are in the horizontal plane and the prismatic joints (the most distal joint of each limb) are in the vertical plane. This orthogonal design was inspired by the design of Ambler [1].

The wheeled robot Marscar is 4 wheeled rover with Ackerman steering.[1]

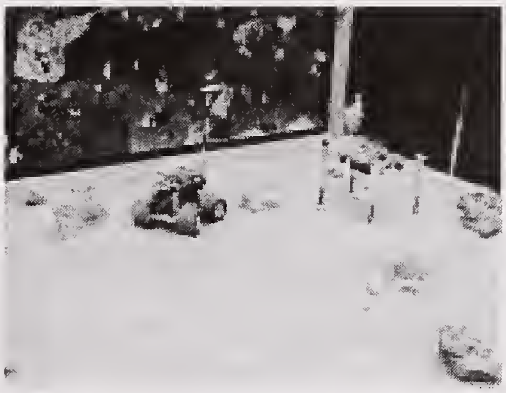


Figure 3: MENO and Marscar in a Simulated Martian Environment

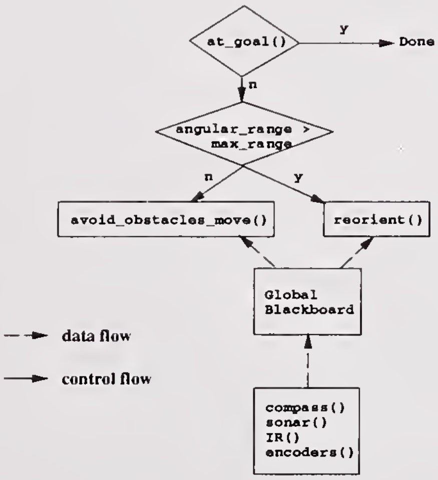
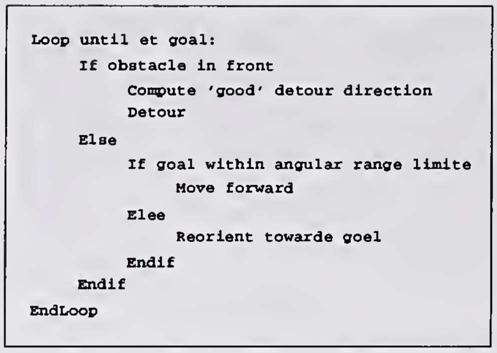


Figure 4: The Control Architecture for the Wheeled Robot

There are two main behaviors that drive both robots. They are `avoid_obstacles_move()` and `reorient_to_goal()`. A schematic of the control architecture is shown in Figure 4.

Onboard computing is all done on a custom board built around a Motorola 68332 microcontroller. A tether is used to supply offboard power for extended testing and for gathering telemetry. The testing is all done in a 3.5 m ×3.5 m sandbox. A single camera suspended 3 m above the center of the sandbox is used for tracking the robot's position. We do *not* use the overhead camera as a source of information for navigation; navigation is done by dead reckoning using information measured by onboard sensors only. The sand surface is nominally flat but not precisely so.

[1] Ackerman steering maintains a particular relationship between the steer angles of the inner and outer wheels in order that the entire robot turn about a single point.



```
Loop until et goal:
    If obstacle in front
        Compute 'good' detour direction
        Detour
    Else
        If goal within angular range limite
            Move forward
        Elee
            Reorient towarde goel
        Endif
    Endif
EndLoop
```

Figure 5: The Navigation Algorithm

3.3 The Navigation Algorithm

Both robots above use the same behavior-based navigation algorithm. There are two[2] basic behaviors; 1. Reorient towards goal and 2. Avoid obstacles. The basic idea is for the robot to keep track of its current position using knowledge of its kinematics and proprioceptive sensors (such as wheel encoders on Marscar and joint angle measurements on MENO). The estimator running on board the robot performs a simple dead-reckoning calculation to estimate position and orientation at every move. The ’avoid obstacles’ behavior is also fairly simple - if an obstacle is seen the robot will attempt to detour around it (while keeping track of its position as mentioned above). If no obstacle is blocking the robot, it will attempt to move towards the goal, re-orienting itself if necessary. The navigation algorithm is reactive. A schematic outline of the algorithm is given in Figure 5.

An interesting part of the detour behavior is the use of global information. When an obstacle is detected the reactive strategy is to backup and turn. The direction of the turn is dependent on the current location of the robot and the commanded goal location in global coordinates. The turn direction that reduces the difference between the robot angle and the desired goal angle θ_g is chosen and executed. A purely local strategy would pick one direction at random but the reactive obstacle avoidance behavior is modified to use some global information viz. the goal position.

We also adapt the angular range during a traverse. The basic observation is that small angular errors when the robot is far away from the goal lead to large position errors later. To avoid this we keep the angular range limits (within which no reorientation is necessary) small when the robot is far away from the goal. These limits are progressively increased as the robot nears the goal.

The experiments were performed in a simulated Mars terrain comprised of a crushed red brick sand mixture. The mixture was spread evenly in a 3.5 m by 3.5 m sandbox to a depth of 0.25 m. The sandbox was populated with rocks of varying size (between 0.04 m and 0.2 m in diameter) to simulate Martian rock distributions. The density of the rocks was equal to the Mars nominal density from the Moore distribution [6].

Since the evaluation functions use probability estimates from numerous mission trials, the experimental protocol consists of many robot traverses from start to goal locations in different instantiations of Mars nominal terrain. There are three main loops. During a particular instantiation a number of trials are performed with different start and goal locations. During the course of each of these trials (as the robot is navigating from start to goal) the offboard computer is monitoring time. When a certain time interval δ_t is reached the overhead vision system images the robot and the image is stored with a timestamp. When the current trial is over the sequence of images taken is postprocessed to extract the (x, y) location of the robot as a function of elapsed time. This information is stored in a file and the next trial begins. The procedure is terminated when all the exemplar start/goal locations have been used in every exemplar terrain. The protocol for energy is exactly the same as the time trials but instead of monitoring the time elapsed, the power draw is monitored. Using this a running total of the energy consumed is maintained. When the energy consumption reaches a threshold δ_e the robot is imaged.

Once the data recording the position and orientation of the robot is obtained using the protocol described above, it is processed to create plots of the required probability estimates that yield the previously defined figures of merit that we are interested in. The data processing steps for the time trials are as follows:

- Fix a given time resource value (t_0)
- Fix a required minimum displacement (r_0)
- Build a plot of $\pi(r > r_0)$ vs. t
  1. for each of the n data sets, $\forall t < t_0$ compute $r = \sqrt{(x - x_s)^2 + (y - y_s)^2}$
  2. a = number of r values greater or equal to r_0
  3. use a/n as the required probability estimate
- Compute nondimensionalized $\tau = \int_0^{t_0} \pi_{r_0}(t)dt$

[2] The legged robot also has balancing and gaiting behaviors at a lower level. They are discussed elsewhere [8]

- Repeat above steps for different values of r_0 and t_0

The data processing steps for the energy trials are similar. In both outlines above (x_s, y_s) is the robot start location and $\pi(r > r_0)$ denotes the probability that the displacement r from the start location is greater than r_0.

4 Data Analysis

The experiments were performed in simulation and with the physical robots. The datasets discussed here thus contain results from both. We will however restrict ourselves to a discussion of the datasets from the physical robots since space constraints do not allow a complete discussion here. The interested reader is referred to [9] for a complete account.

4.1 Mobility Trials and Clustering in Tradeoff Space

The first step in calculating the figures of merit is to calculate the probability of reaching k robot lengths as functions of time and energy. Since we have multiple trials we estimate this probability as the fraction of trials in which the displacement was greater than kl as functions of time and energy consumption. In Figure 6 the probability of Marscar reaching the threshold displacement kl is shown for various values of k. The quantity l is intended to be a measure (with dimensions of length) of the robot size. We use the cube root of the volume of the smallest rectangular box in which the robot can be packed. For Marscar $l = 0.35$ m. All the trials were done in Mars nominal distributions. One can see a reasonable agreement between the simulated dataset and the dataset collected from the physical robot. The simulated dataset consisted of 200 trials and the physical dataset consisted of 40 trials. The probability estimates of the simulated dataset are smoother compared to the physical dataset due to the larger sample size. The general behavior of the family of curves shown in Figure 6 is a monotonic rise to saturation. The interpretation of these curves is the likelihood of success (at navigating through the obstacle field) as a function of the available resource (time). A higher k value corresponds to a longer traverse and thus involves greater ability in penetrating obstacle fields. As k is increased for the same robot the probability of achieving the same degree of success decreases.

In Figure 7 a family of curves is shown which plot the probability of Marscar achieving a threshold displacement kl as a function of energy consumed. As in

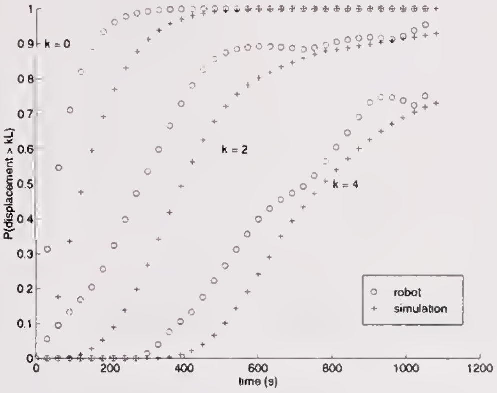


Figure 6: Marscar - Probability of reaching threshold displacements vs. time in Mars nominal terrain

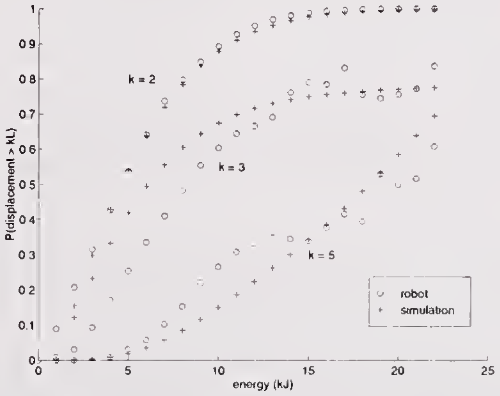


Figure 7: Marscar - Probability of reaching threshold displacements vs. energy in Mars nominal terrain

the case of the plots in the previous figure, the probability of greater success shows an asymptotic rise to saturation. Figure 7 shows the probability estimates for the simulated as well as physical datasets. As one can see there is a good match between the two. As in the previous case larger k values imply longer missions and thus are harder to achieve for the same value of the energy resource. Performance degrades as k is increased. As in the time trials with Marscar, the physical datasets in Figure 7 are the result of 40 trials and the simulated datasets are the result of 200 trials.

In order to compute the figures of merit for MENO in Mars nominal terrain we follow the same data analysis procedure as before. The curves showing the plots of the probabilities of achieving the threshold displacement kl as a function of time elapsed are shown in Figure 8. As in the previous cases increasing values of k signify longer missions. For MENO $l = 0.47$ m. The

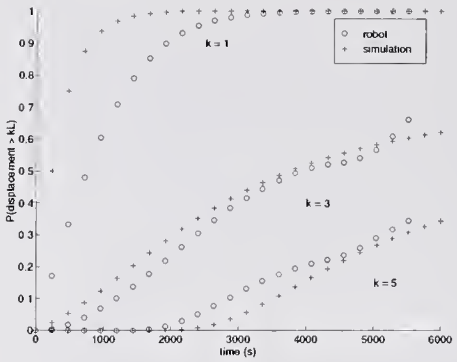


Figure 8: MENO - Probability of reaching threshold displacements vs. time in Mars nominal terrain

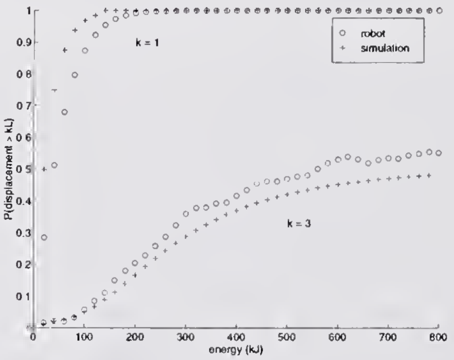


Figure 9: MENO - Probability of reaching threshold displacements vs. energy in Mars nominal terrain

physical datasets shown in Figure 8 were computed using 40 trials in Mars nominal terrain and the simulated datasets were generated using 200 trials in simulation.

The last datasets of interest in the current series are the behavior of MENO as a function of the energy consumed in Mars nominal terrain. The relevant plots are shown in Figure 9.

In the notation of Chapter 4 we now have plots of $\pi_{kl}(t)$ and $\pi_{kl}(e)$; the probabilities of the achieving certain threshold displacements as functions of time and energy. Using $t_0 = 40$ min and $e_0 = 200$ kJ as representative numbers for the mission under study we calculate the two figures of merit using Equations 1 and 2 for different values of k. These values are shown in Table 1.

Figure 10 shows the τ and η values for the two robots in the tradeoff space. The lower left hand side of the plot (signifying lower evaluation scores) is the space occupied by the legged robot. The wheeled system has better scores on both time and energy axes. The evaluation functions are evaluated for 4 different values of k. Irrespective of the k value the wheeled robot outperforms the legged robot. The functions η and τ thus partition the design space.

Table 1: The Figures of Merit for MENO and Marscar for Different Traverse Lengths

k	Marscar		MENO	
	τ	η	τ	η
2	14.7	11504	2.7	6911
4	12.1	11429	2.1	5957
6	9.4	9718	1.1	3609
8	7.3	9635	0.8	2745

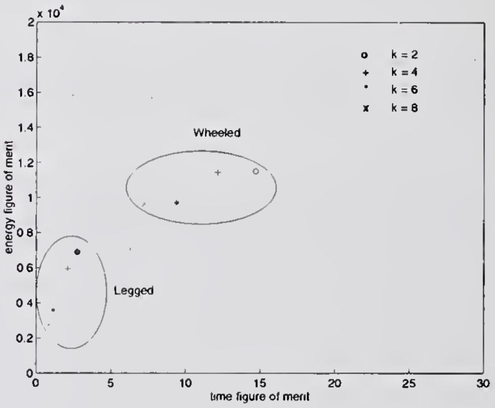


Figure 10: A Comparison of MENO and Marscar in Mars Nominal terrain for Different values of k

To illustrate the cause of the difference in the evaluation scores it is useful to re-examine Figures 6 and 7 on the same scale. This is done in Figure 11 where we show the probability estimates for both MENO and Marscar with $k = 5$ as a function of the time elapsed. Seen on the same axis it is obvious that the wheeled system does better with the 'area under the curve' metric since it is a lot faster than the legged system in this terrain (the Mars nominal rock distribution).

If Figures 8 and 9 are plotted on the same axis a similar conclusion can be drawn regarding the energy

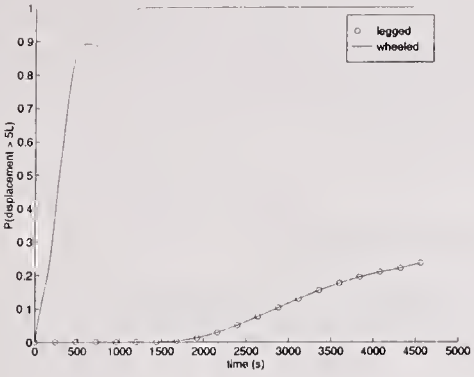


Figure 11: A Comparison of MENO and Marscar in Mars Nominal terrain for $k = 5$ as a Function of Time Elapsed

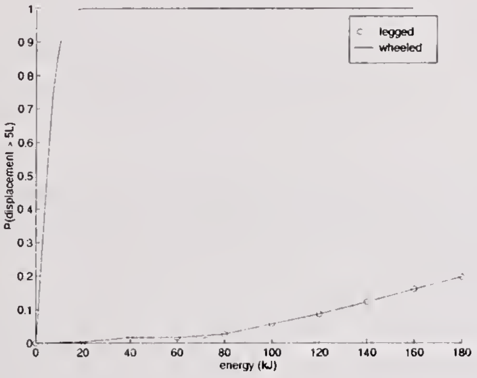


Figure 12: A Comparison of MENO and Marscar in Mars Nominal terrain for $k = 5$ as a Function of Energy Consumed

scores. This is shown (again for $k = 5$) in Figure 12. The wheeled robot needs far less energy to cover the same distance compared to the energy consumption of the legged robot over a similar distance for this particular rock distribution.

4.2 Sensitivity Studies - Environment

One of the objectives of this study was to measure the effects of changes in environmental parameters on the mobility metrics. The environment model used in this study is the distribution of rocks called the Moore distribution. In the vicinity of a previous mission to Mars (the Viking II mission) the density of rocks is much higher than the Mars nominal distribution used thus far. The effect of terrain clutter is very clearly seen in the two metrics. In the case of both robots, increased clutter leads to performance degradation. However it is

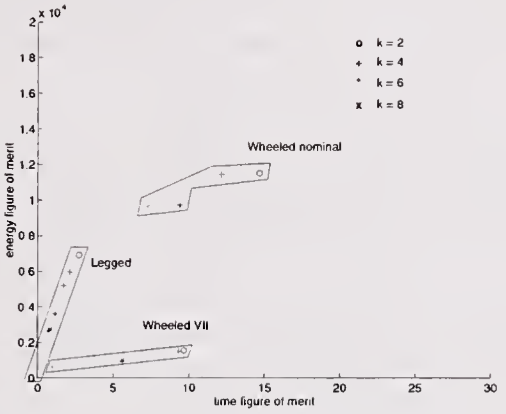


Figure 13: MENO and Marscar mobility in Viking II (cluttered) terrain for different values of k

interesting to note that the wheeled system is affected far more than the legged system. This is largely due to the fact that the increased clutter leads to significantly longer paths for the wheeled system whereas the legged system is able to go over many more obstacles and even though it is slower its performance is comparable to the legged robot. This is shown in Figure 13.

As one can see in Figure 13 the Marscar cluster moves dramatically to the left and down when the terrain was changed from Mars nominal to Viking II. MENO performance also suffered as seen in Figure 13 but not as dramatically. For this environment, its energy figure of merit is better than Marscar.

4.3 Scalarization of the Metrics

The metrics τ and η can be combined into a single scalar metric using a weighted linear combination. From the data presented in this Chapter we see that the wheeled robot outperforms the legged vehicle along both dimensions in Mars nominal terrain. The scalarization chosen should preserve this ordering. A standard technique is to use a weighting function which is either linear or quadratic and maximize the combination of the two metrics. However the problem of how to choose the weights still remains. Instead of an ad hoc solution we use domain knowledge to postulate a feasible scalarization technique.

On one axis (τ) we are measuring the robot's ability to use time effectively and on the other (η) we measure effective energy utilization. The fundamental unit of conversion between them is the maximum power delivered by the onboard power source. If the power source is capable of delivering α W then we weight energy and time in the ratio $1 : \alpha$.

We computed the scalarized scores for $k = 6$ for the different cases reported in this Chapter using $\alpha_1 = 30$, $\alpha_2 = 40$ and $\alpha_3 = 50$. Using this scalarization technique it is clearer that in sparse obstacle distributions the legged system should be the preferred design while in dense obstacle distributions (such as the Viking II site) the nominal configuration of the legged robot MENO is the better design using these metrics and this particular linear scalarization.

5 Discussion

Values of the two metrics, τ and η for Marscar are significantly superior to the MENO values. The effect of obstacle clutter, though, is more pronounced on the wheeled robot.

There are three interesting aspects of the data presented here which form the basis for substantial future research. The first deals with the following design question: "In what parts of the design space are good designs found ?". At first glance it may seem like the answer is obvious - by definition it would seem like the designs leading to the highest values of the evaluation functions are the good parts of the design space. However, a closer look suggests that the real 'sweet spots' in the design space are those where the design is insensitive to changes in the environment. For example, MENO in its nominal configuration is insensitive to changes in rock density. If there is large variability in the expected terrain density it may be a better decision to pick a design like MENO even though it has low evaluation scores compared to other designs. We are thus led to believe that future scalarization efforts should include weighted contributions from select components of the evaluation gradient in addition to the values of the evaluation functions themselves.

The second interesting point also concerns the evaluation gradient. Locations in the design space where the evaluation gradient becomes very large also provide interesting insight into design methodology. We suggest that these locations in the design space signal a 'breakdown' in the current kinematic design and a discrete jump to a new structure is indicated (with higher articulation perhaps or with a larger number of wheels).

A third application of the metrics proposed here is to global optimization. While the technique for extrapolating performance shown here is local, it is possible to extend it by instantiating a chain of local models and following the evaluation gradient to an optimal set of parameter values.

6 Acknowledgments

The author gratefully acknowledges help from George Bekey in the development of these ideas. The author is partially supported by two DARPA programs; the Tactical Mobile Robotics (TMR) program under contracts F04701-97-C-0021 and DAAE07-98-C-L028, and the Mobile Autonomous Robot Software (MARS) program under grant DABT63-99-1-0015. The case studies described here were performed under the auspices of NASA grant #959816 from Caltech/JPL.

References

[1] J. Bares and W. Whittaker. Configuration of autonomous walkers for extreme terrain. *International Journal of Robotics Research*, 12(6):535–559, 1993.

[2] K. R. Lietzau. Mars micro rover performance measurement and testing. Master's thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, December 1993.

[3] S. Mahalingam and W. L. Whittaker. Terrain adaptive gaits for walkers with completely overlapping leg workspaces. In *Proc. Robots 13*, May 1989.

[4] R. B. McGhee and A. A. Frank. On the stability properties of quadruped creeping gaits. *Mathematical Biosciences*, 3(3/4):331–351, 1968.

[5] D. A. Messuri and C. A. Klein. Automatic body regulation for maintaining stability of a legged vehicle during rough terrain locomotion. *IEEE Journal of Robotics and Automation*, RA-1(3):132–141, 1985.

[6] H. J. Moore and B. M. Jakosky. Viking landing sites, remote-sensing observations and physical properties of martian surface materials. *Icarus*, 81:164–184, 1989.

[7] P. Nagy, S. Desa, and W. L. Whittaker. Energy-based stability measures for reliable locomotion of statically stable walkers: Theory and application. *The International Journal of Robotics Research*, 13(3):272–287, June 1994.

[8] G. S. Sukhatme. The design and control of a prototype quadruped microrover. *Autonomous Robots*, 4(2):211–220, April 1997.

[9] G. S. Sukhatme. *On the Evaluation of Autonomous Mobile Robots*. PhD thesis, University of Southern California, May 1997.

[10] B. Wilcox. Mobility characteristic curve. Jet Propulsion Labs, IOM 3472-91-019, 1991.

Search Graph Formation for Minimizing the Complexity of Planning

Alberto Lacaze

Computational Intelligence Laboratory, ECE
University of Maryland, College Park

Stephen Balakirsky

Intelligent Systems Division
National Institute of Standards and Technology

Abstract

A large number of path planning problems are solved by the use of graph based search algorithms. There are a variety of techniques available to optimize the search within these graphs as well as thorough studies of the complexity involved in searching through them. However, little effort has been dedicated to constructing the graphs so that the results of searching will be optimized.

The commonly used approach for the evaluation of complexity assumes that the complexity of a path planner can be evaluated by the number of nodes in the graph. However, in many path planning problems (especially in complex, dynamic environments) the evaluation of the cost of traversing edges is the major culprit of computational complexity. In this paper we will assume that the complexity associated with the computation of cost of traversing an edge is significantly larger than the overhead of searching through the graph. This assumption creates non-trivial complexity results that allows to optimize the creation of the graph based on the computational power available.

We will present a numerical evaluation of several graph creation algorithms including the commonly used four and eight connected grid. Different scenarios for which ground truth is available are explored. Comparison among the graph creation algorithms reveals serious downfalls that are common practice throughout the literature.

1 Introduction

Planning can be defined as the process of finding the steps necessary to bring a system from an initial (current) state to a final (desired) state. Most planning techniques represent the planning problem in a graph $G(V, E)$. Where V is a set of vertices, and E is a binary relation on V [6, 7, 9]. The elements of the set V are called vertices and represent states. The elements of the set E are called edges and represent the ability of the system to move from one state to another. In planning graphs, the edges are ordered or unordered pairs of vertices, (v_i, v_j) where $v_i \in V$ and $v_j \in V$. A walk is an alternating sequence of vertices and edges, a trail is a walk with distinct edges, and a path is a trail with distinct vertices.

When solving a planning problem, we must find a path or plan from a starting vertex v_s to an ending vertex v_e while minimizing a cost function $C = \sum_s^e w_{ij}$ where w_{ij} is the cost of traversing the edge (v_i, v_j). Some planning problems can be solved by algorithms with polynomial complexity. Unfortunately, these tractable set of problems covers only a few of the relevant problems encountered in path planning. Most problems, however, can only be solved by polynomial algorithms on non deterministic machines, ie NP. For a thorough study on the problem of tractability and its taxonomy see [8].

One very useful tool when fighting the computational complexity of planning is the creation of hierarchies of planners. The Real-time Control System (RCS) reference model architecture is one such architecture and it has been successfully applied to multiple diverse systems [1, 3]. The target systems for RCS are in general, complex control problems. Although it has been shown [2, 10] that the complexity of a control problem is reduced by the use of a hierarchical control system, the reduction of error as a function of complexity at one level of the hierarchy has been mostly overlooked.

The complexity of search algorithms inside a graph has been thoroughly studied [11, 13, 14]. However, with few exceptions [4, 12], little attention has been paid on how the graph should be built with some exceptions [4, 12]. In most cases, it is recommended that the graph for search on "empty space" should be built using grids, Voronoi diagrams, or visibility graphs. It

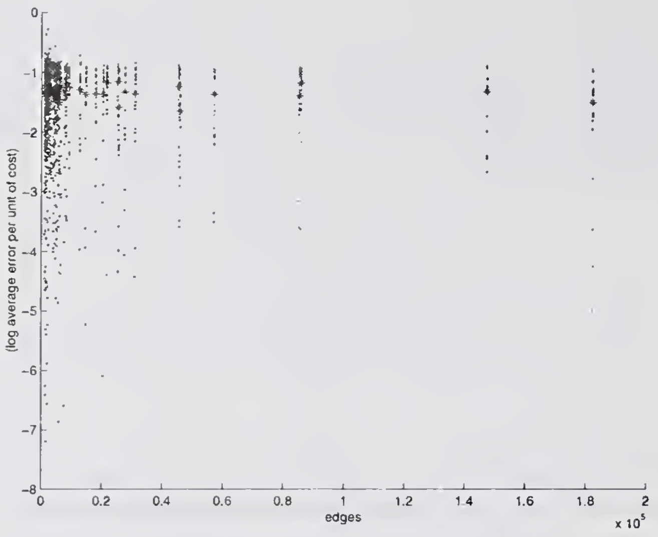


Figure 1: Average error for a 4 connected grid.

is not clear from the literature which of these methods should be used and when. Moreover, in most cases the complexity of algorithms is calculated solely based on the number of vertices in the graph. In most path planning problems, the computational complexity of calculating the cost of the edges is orders of magnitude higher than the actual time spent searching through the graph once these values have been calculated.

2 Numerical Exploration of Graph Creation

In order to compare the different graph formation algorithms, we started by defining a simple test scenario. The analytical closed form evaluation of the complexity of finding the optimum path taking under consideration the placement of the vertices in the solution space becomes easily intractable. Therefore, we decided to study the problem numerically. In the experiments presented in this paper, simple Euclidean distances were used to calculate the cost of traversing the edges. The advantage of using this measure is that we have ground truth. We assumed that the Euclidean distance is calculated with an accuracy of five significant figures.

2.1 Grid Based Graphs

By far, the most commonly used graph for search in planning algorithms is the four-connected square grid. In this kind of graph, the vertices are placed at regular intervals and it is assumed that each vertex is connected to four (or eight) of its closest neighbors.

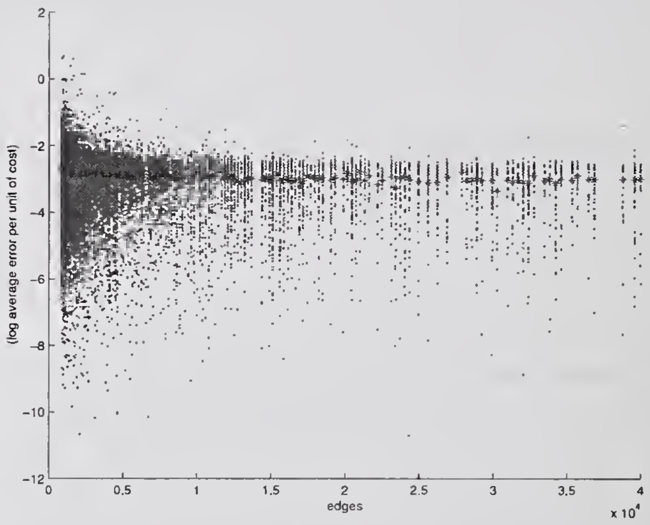


Figure 2: Average error for a 8 connected grid.

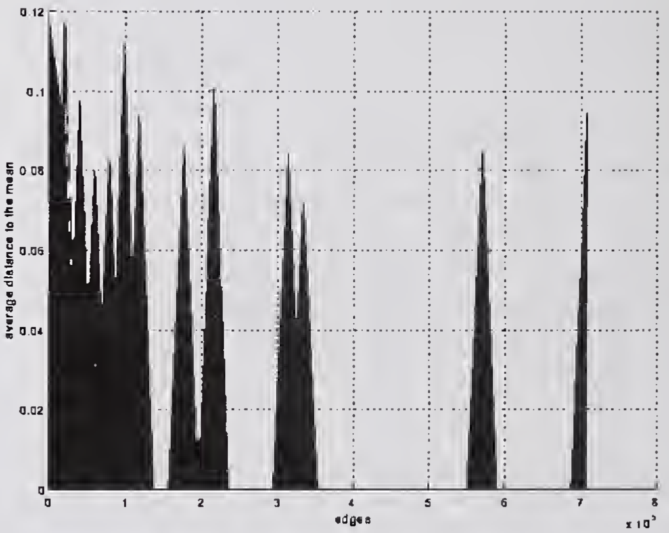


Figure 3: Average distance to the mean.

We built a two dimensional four-connected square grid with a random number of vertices. We repeated this experiment several times. Figure 1 shows $log(error)$ where $error$ is defined as

$$error = abs(d_{s,e} - ((\sum_{vs}^{ve} d_{i,i+1}) + d_{s,v_s} + d_{e,v_e})) \quad (1)$$

s is a randomly selected starting point, e is a randomly selected ending point, v_e is the closest vertex in the graph to e, v_s is the closest vertex in the graph to s, $d(i,j)$ is the Euclidean distance between two points. Please note that this cost function may underestimate the real error of traversing the planned graph as it is assuming that d_{s,v_s} and d_{e,v_e} are Euclidean. This is a best case scenario.

The summation in the equation represents the added cost of the optimal path through the graph. The average error (marked with a black star in the Figure) is kept constant as the number of edges is changed. The different values at a particular number of edges correspond to the different number of times that the experiment was performed using different e and s.

Figure 2 shows the error function shown in 1 applied to a eight-connected grid. As expected, the error function settles at a lower error. By comparing the 4-connected grid to the 8-connected grid we can appreciate that the average error decreases with the higher connectivity, however in both cases, the error quickly settles to a constant value.

Please note that in both cases, increasing the number of edges, and therefore increasing the computational complexity gives us very modest improvements of the final cost. Another problem found experimentally with the 4 and 8 connected grids using this cost function is that there are many paths that have exactly the optimal cost. This has the effect that the optimal path that the algorithm will choose, may wander off the "expected" straight path line from e to s. In other words, many paths within the parallelogram defined by v_s and v_e have exactly the same "optimal" cost. Another effect that results from square grids is that the error varies significantly depending on the direction of travel. A numerical evaluation of this deviation can be appreciated by examining Figure 3. The large average distance to the mean is due to the fact that some s and e happened to be horizontal or vertical, therefore giving small error, while some created a very costly stair-step paths through the graph.

2.2 Shaking the Grid

Some of the pitfalls of the grid based graphs can be avoided by:

1. Shaking the vertices within the grid. In other words, building a square grid, adding a random displacement to the vertices, and finally connecting all the vertices that are within a neighborhood. The size of the neighborhood dictates the vertices to edges ratio. This has two effects:

   (a) Break the ties among optimal paths so that only one path is found to be optimal. This is very helpful in re-planning systems as it forces to commit instead of randomly flipping among the set of "optimal" paths.

   (b) Create a more uniformly distributed set of vertices where all " directionalities" are represented.

2. Create higher connectivity rates (higher than in the 8-connected grid).

Figure 4 through Figure 7 shows the results of a set of experiments run using the above principles. To compute these figures, the vertices of the grid are placed first in a grid pattern where each point is l apart from its closest neighbor. Next, a random vector is added to each vertex of maximum amplitude $3l$. All vertices within a distance threshold are then connected. By varying the connection threshold, different ratios between the number of nodes and the number of edges are achieved. We can see from Figure 4 that the error decreases as the number of edges increases, approaching the 10e-5 mark set by the 5 significant figures used to calculate the Euclidean distances. Figure 5 shows a top view of the same numerically found error. We can see that even a simple Euclidean cost function creates ripple effects in the final cost.

If we take the assumption that the computational complexity is directly proportional to the number of edges (as it is in most cases), we can see in Figure 8 the error function as a function of the number of nodes. The almost counter-intuitive results can be explained from the fact that by increasing the number of vertices the average cost of an edge decreases. In Figure 9 we assumed that we could only calculate the cost of 40000 edges. By visual inspection of Figure 9 we can determine that the least error is given by about 2000 vertices, and therefore creating a graph where each vertex has 20 connected neighbors.

3 Vehicle Planner Example

In order to validate the above rules of thumb, several experiments were conducted using the Demo III

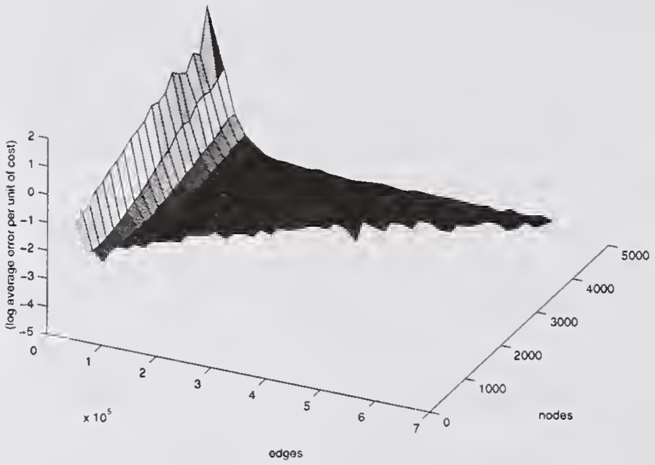


Figure 4: Average error in a shaken grid.

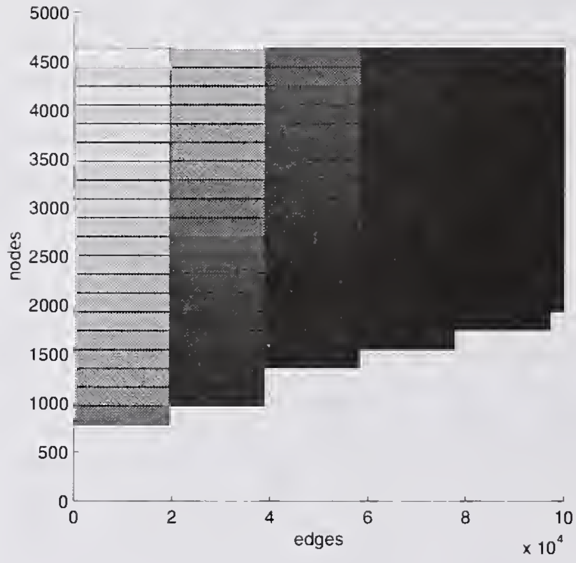


Figure 5: Average error in a shaken grid:top view.

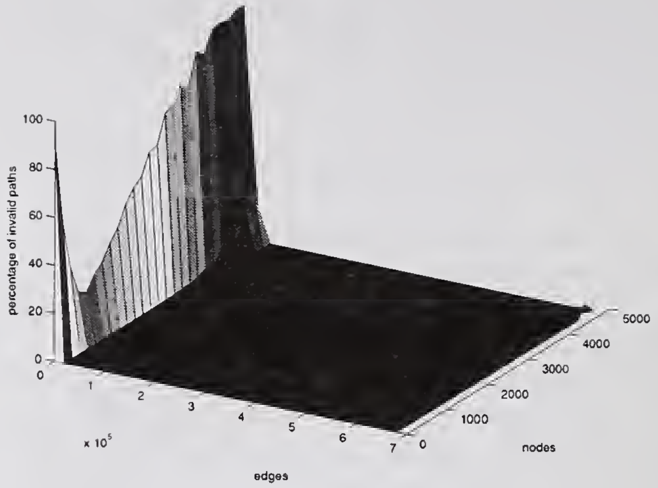


Figure 6: Percentage of failed planning processes in shaken grid.

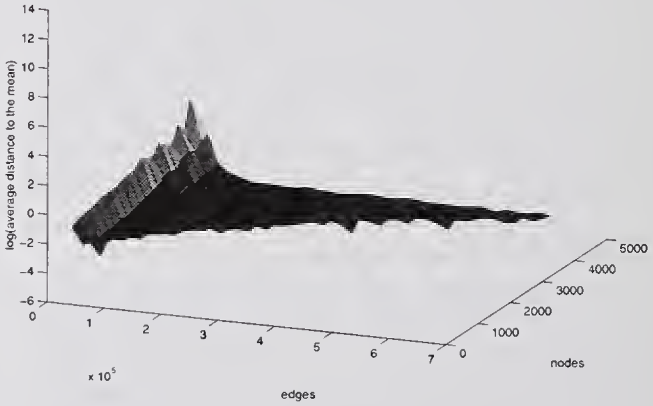


Figure 7: Average distance to the mean in shaken grid.

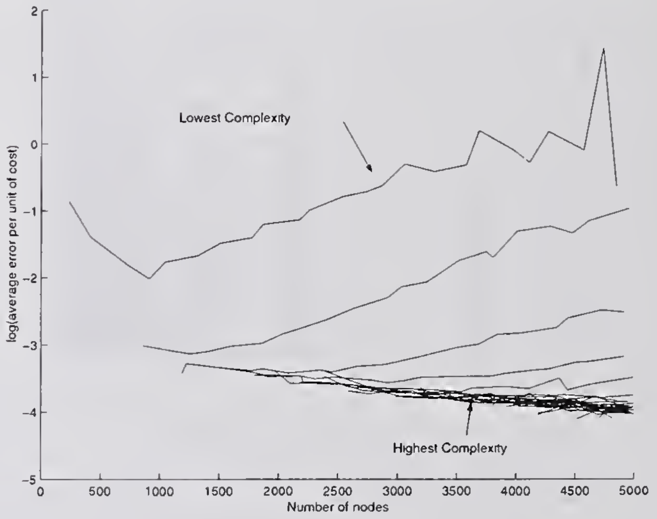


Figure 8: Error for different complexities and varying number of vertices

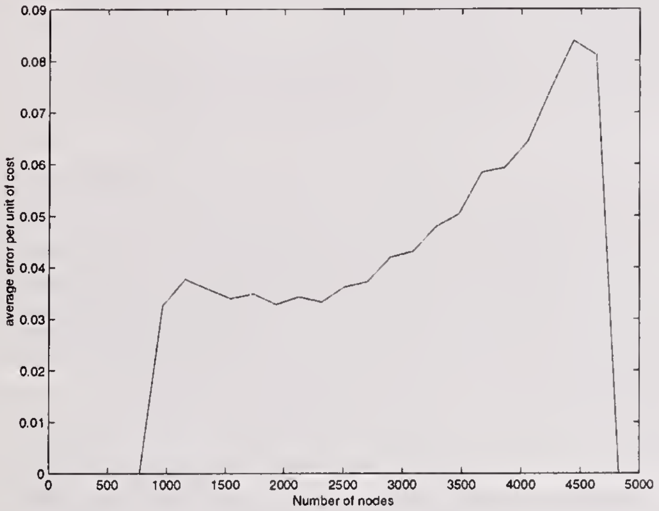


Figure 9: Error for fixed complexity and varying number of vertices

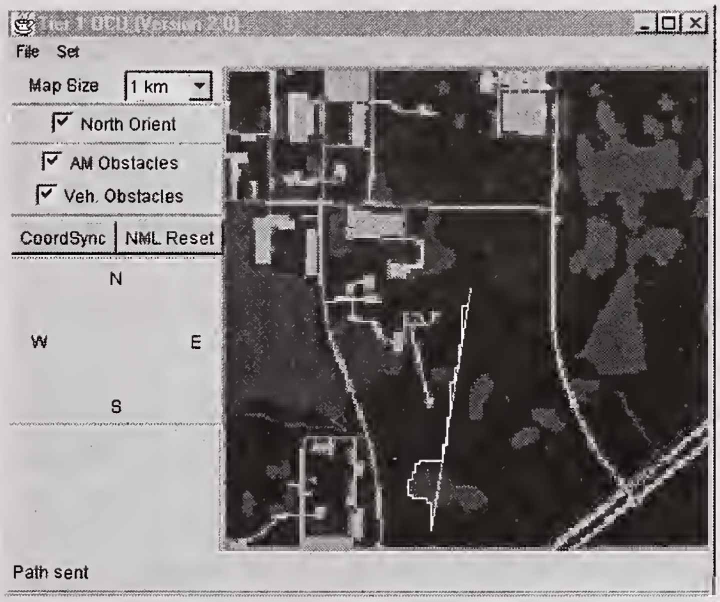


Figure 10: Planning result for complex cost map with four-connected graph.

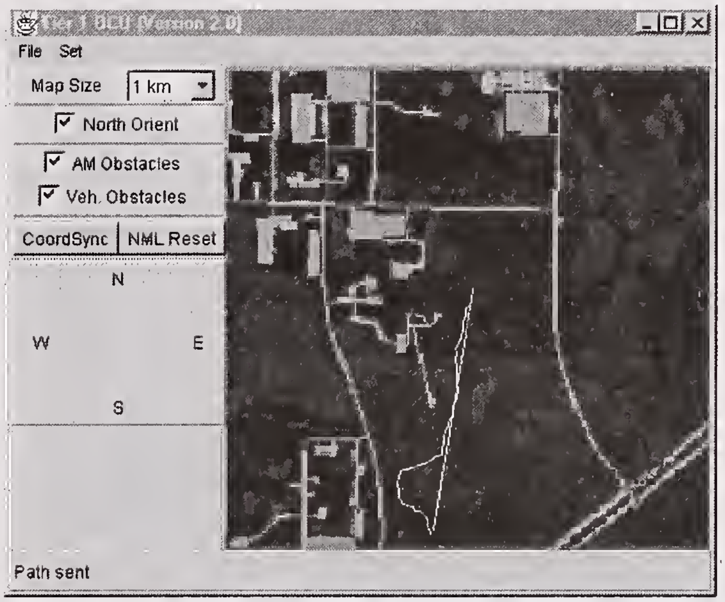


Figure 11: Planning result for complex cost map with many-connected graph.

Vehicle Level Planner [5]. In these experiments, a four-connected graph and a shaken graph of the form of section 2.2 were run using a complex world model and cost function. The four-connected graph had a grid size of 8 meters with 61012 connections and the shaken graph had a grid size of 11 meters with 45086 connections (26% fewer connections) and was shaken ±5.5 meters. The world model contained a priori information on the NIST grounds at 4 meter resolution including the locations of wooded areas, buildings, roads, and fences. It should be noted that the world model resolution is twice that of the four-connected graph and almost three times that of the highly-connected graph.

In the Demo III Vehicle Level Planner, the planning module passes path segment endpoints (the vertices of the planning graph) to the world model for evaluation. The world model simulates driving a straight line path (the edges of the planning graph) between these end points and returns the cost of traversal to the planner. The planner then conducts an optimal search algorithm to find the cheapest path (in reference to the cost function used by the world model). The cost function used by the world model favored paths that avoided roads and buildings, and drove next to, but not in wooded areas combined with the time of traversal of the route (assumed uniform vehicle velocity over the route segment).

The straight line segments used by the world model may cause plan failures when the resolution of the planning graph is less then that of the world model.

This occurs when a very narrow low-cost corridor is surrounded by a very high cost area. It may occur that there are no straight line segments at the graph resolution that traverse this low-cost corridor. This phenomenon can be avoided in the highly-connected graph by adding additional vertices in these high payoff areas. This approach was not taken in the experiments described below.

Using this planning system, we found that the highly-connected graph performed as much as 27% better then the four-connected graph, even though it used 26% fewer connections. Sample output paths may be seen in Figure 10 for the four-connected graph and Figure 11 for the highly-connected graph. A snap-shot of the world model may be seen as the background of these images. As one would expect, the benefit of using the highly-connected graph is directly tied to the shape of the optimal path. For straight paths, the two graphs performed on par with each other. For paths which required many turns, the highly-connected graph significantly outperformed the four-connected graph.

4 Conclusion

- "Optimal" paths found using the four-connected grid based graph are in general, directionally biased, favoring the traversal of the space in certain directions and not in others. They also create symmetries that result in noncommittal paths. Shaken grids and high connectivity between vertices was shown numerically to improve these pitfalls.

- The number of edges in the graph and their cost evaluation are in most cases, the major culprit for computational complexity. Therefore, it is recommended that the graph design process starts by determining the number of edges that can be evaluated, and then selecting the number of vertices that give the least error.

- Numerical evaluation of the error are in most cases the only way to select parameters for the formation of search graphs in complex environments. Most analytical evaluations of the complexity in the literature make the assumption that the burden of computational complexity is in the "opening" of the vertices in the search graph, and are not readily applicable to planning problems.

References

[1] J. Albus. Outline for a theory of intelligence. *IEEE Transactions on Systems, Man, and Cybernetics*, 21:473–509, 1991.

[2] J. Albus, A. Meystel, and A. Lacaze. Multiresolutional planning with minimum complexity. *Intelligent System and Semiotics*, 97.

[3] J.S. Albus. *Brain, Behavior, and Robotics.* McGraw-Hill, 1981.

[4] R.S. Alexander and N.C. Rowe. Path planning by optimal-path-map construction for homogenous-cost two-dimensional regions. In *IEEE International Conference on Robotics and Automation - 1990*, 1990.

[5] Stephen Balakirsky and Alberto Lacaze. World modeling and behavior generation for autonomous ground vehicles. In *Proceedings IEEE International Conference on Robotics and Automation*, 2000.

[6] J. Bondy and U. Murty. *Graph Theory with Applications.* North Holland, 1976.

[7] N. Deo. *Graph Theory with Applications to Engineering and Computer Science.* Prentice Hall, 1974.

[8] M. Garey and D. Johnson. *Computers and Intractability.* Freeman, 1979.

[9] F. Harary. *Graph Theory.* Addison Wesley, 1972.

[10] Y. Maximov and A. Meystel. Optimum design of multiresolutional hierarchical control systems. In *Proceedings of IEEE Int'l Symposium on Intelligent Control*, pages 514–520, Glasgow, U.K., 1992.

[11] C.H. Papadimitriou. *Computational Complexity.* Addison Wesley, 1994.

[12] F.P. Preparata and M.I. Shamos. *Computational Geometry, An Introduction.* Springer Verlag, 1988.

[13] J.H. Reif. Complexity of the generalized moving problem. In et al Schwartz, editor, *Planning Geometry and Complexity of Robot Motion.* Ablex Publishing Corporation, 1987.

[14] J.T. Schwartz, M. Sharir, and J. Hopcroft, editors. *Complexity of the Generalized Moving Problem.* Ablex Publishing Corporation, 1987.

Metrics for Embedded Collaborative Intelligent Systems[1]

Michael E. Cleary, Mark Abramson, Milton B. Adams, Stephan Kolitz
Draper Laboratory
555 Technology Square, MS 3F, Cambridge, MA 02139
{mcleary, mabramson, adamsm, kolitz}@draper.com

ABSTRACT

The intelligence of a network of agents is reflected in the complexity of missions that can be accomplished, the degree of coordination/cooperation among the agents, and the level of uncertainty the system can tolerate and still accomplish its missions. The networked system must be able to evaluate a situation, devise an appropriate response, and act accordingly. Metrics must be devised to capture the complexity and surprises of the real world, and to capture the system's need to reason about its situation so as to uncover unanticipated problems and opportunities. Inputs for developing autonomous capability specifications (and thus metrics of interest) include (1) descriptions of expected missions, (2) the space of mission parameters, and (3) the cost/benefitratio for operational concepts. These inputs come from both current and anticipated missions. Several of our recent projects have sought to quantify operational metrics for autonomous ground, air and undersea vehicles. This paper presents our approach to high-level design of autonomous vehicles that produces the three inputs for metric development. The approach and parameter spaces are illustrated with examples derived from several vehicle projects.

Keywords: metrics, intelligence quotient, intelligent systems, autonomous systems, collaborative systems, situation awareness, planning under uncertainty, orders of intelligence.

1 Introduction

The intelligence of a network of agents is a complex characteristic that can be quantified and measured in a wide variety of ways. Our work on the design of intelligent autonomous vehicles and programs to develop such vehicles has made clear that the type of metric we develop will be chosen to meet a particular objective. For instance, commercial sponsors will likely optimize some functionality, while researchers may try to optimize some measure of "pure" intelligence. After reviewing a number of systems in ground, air and undersea domains, it becomes clear that the .major characteristics of intelligence for any complex set of vehicles are the broad areas of multi-vehicle collaboration, understanding the world they operate in (situation awareness) and responding appropriately (planning under uncertainty).

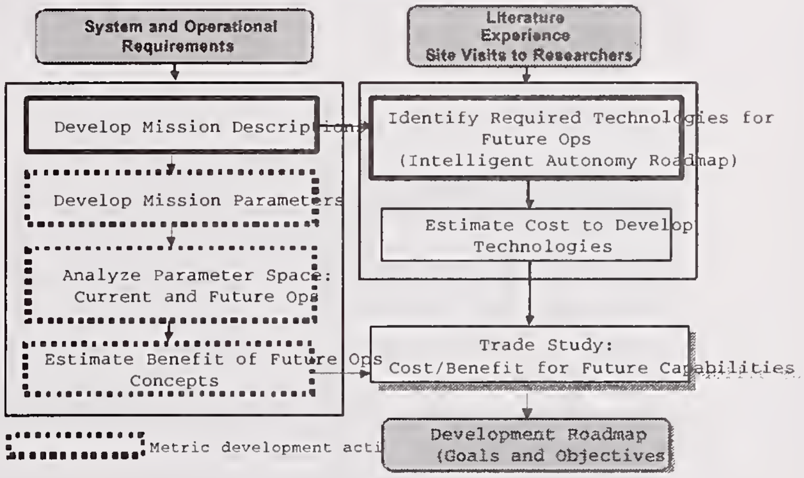

System and Operational Requirements
Literature
Experience
Site Visits to Researchers
Develop Mission Descriptions
Identify Required Technologies for Future Ops
(Intelligent Autonomy Roadmap)
Develop Mission Parameters
Estimate Cost to Develop Technologies
Analyze Parameter Space: Current and Future Ops
Estimate Benefit of Future Ops Concepts
Trade Study: Cost/Benefit for Future Capabilities
Metric development acti
Development Roadmap (Goals and Objectives

Figure 1 -- Development Roadmap

To guide the design of intelligent vehicles for particular domains, we have used the process illustrated in Figure 1. There are two major efforts shown – the left column focuses on the missions the vehicle is intended to accomplish, while the right column focuses on the technologies required to accomplish those missions. The two columns could be loosely labeled requirements pull and technology push, respectively. The areas we have considered for metric analysis to date are those shown surrounded with dotted lines. Once thorough descriptions of the vehicles' missions are developed, those are reviewed to extract parameters that affect performance. The mission descriptions are then extended to probe the space of the identified parameters. This process is illustrated in detail in Section 2.

A more humanly intuitive representation of the parameter space was sought, since the bare listing of parameters can be daunting (Section 3). This introduces significant subjectivity, but allows aspects of intelligence to be clustered that seem to lead to strong collaborative systems.

Section 4 discusses an attempt to quantize intelligence into "orders" of intelligence. It begins with the point that

[1] Copyright 2000 The Charles Stark Draper Laboratory, Inc.

"intelligence" is still a relatively undefined area, needing substantial work in the component technologies and in the development of appropriate metrics. Despite that reservation, candidate levels of intelligence capability are described that might serve as an IQ for autonomous systems.

Costs are another aspect of intelligence that require attention and metrics (Section 5). For instance, a sponsor may seek to develop a comprehensive technology roadmap that will determine what technologies need investment to meet a particular set of system and operational requirements.

The paper concludes with a brief discussion of some future directions for our work (section 6) and a summary (section 7).

2 Construction of Parameter Space

The simplest way to evaluate a system's success or failure at its task is often binary – did it accomplish some goal? For instance, in RoboCup Soccer [3] as in human games, a single score is the final arbiter of success. However, the single score does not capture the complexity of the domain or of the team's approach to various elements of the problem. Thus additional "scores" are developed that rate game players on the skills that contribute to the final game score. Such more detailed scores can be combined into a single weighted score, using multi-objective optimization techniques [1,2]. However, that requires significant work to determine appropriate weightings and combination techniques.

The first step toward such a development is to flesh out the parameter space of the task. A large number of factors can be considered in a thorough analysis of a collaborative group of vehicles. We use the three characteristic areas named above (collaboration, situation awareness, and planning under uncertainty). The following incomplete lists indicate some of the important elements for robots facing dangerous situations (military or other). Each metric on the list requires a range of acceptable values and a weighting factor for combining them with other components. The factors can then be processed to produce a combined metric if such a score is desired.

- **Multi-vehicle collaboration factors**
  - number of interacting agents
  - degree of coordination/cooperation among the agents
  - degree of improvement in situation awareness due to multiple vehicles
  - success of dynamic replanning to maintain configuration for communication
- **Situation awareness**
  - amount of complexity and surprise of real world captured
  - number of elements
  - level of interactions between elements
  - dynamism
  - model complexity for target identification
  - observability
  - environmental challenge
    - + clear air/daylight – to – storms at night
    - + desert (all is visible) – to – mountainous (hard to see details)
    - + textured (landmarks differ) – to – desert/no texture
  - threat types
    - + from known type/location – to – suspected – to – unknown till aggression
    - + from id is straightforward (e.g., surface-to-air-missile (SAM) radar) – to – difficult/uncertain (visual or synthetic aperture radar (SAR), near friendlies, signature similar to neutral or friendly
  - neutrals
    - + known type/location – to – threats masquerading as neutrals
  - friendlies
    - + known type/location – to – identify-friend-foe (IFF) transponders off/broken or known but near threats
  - navigation
    - + sensors functioning and low uncertainty – to – sensors dropping out/damaged or high uncertainty
  - vehicle state (including equipage)
    - + sophistication of health monitoring and reconfiguration
  - time to sense and assimilate (separate from time to plan)
    - + enough time – to – insufficient time due to tempo or number of targets (so need to prioritize sensing and assimilation)
  - can successfully identify a target
  - can detect environmental changes of the following types:
    - + threats
    - + terrain
    - + collision
    - + targets of opportunity
- **Decision making and executing under uncertainty**
  - extent that system reasons about its situation
    - + uncovers unanticipated problems
    - + uncovers opportunities
  - level of uncertainty the system can tolerate
  - performs under available time to plan

- dynamic time constant that system can reason within
- stochasticity - number of contingencies handled by system
- number of decisions (i.e., size of planning problem)
- quality of plan generation / selection algorithms
- quality of planning approach (algorithms and representations)
- complexity of mission / problem
- complexity of controllable system
- number of plan elements in flux simultaneously
- number of levels in planning problem
- ability to perform dynamic replanning due to:
  + change in mission objectives
  + environmental change detected

Such a list of parameters is daunting, and only becomes more difficult to grasp and synthesize as the level of detail grows. A more intuitive representation was sought to support analysis of the trade-cffs involved in system design and funding. The result is discussed in the following section.

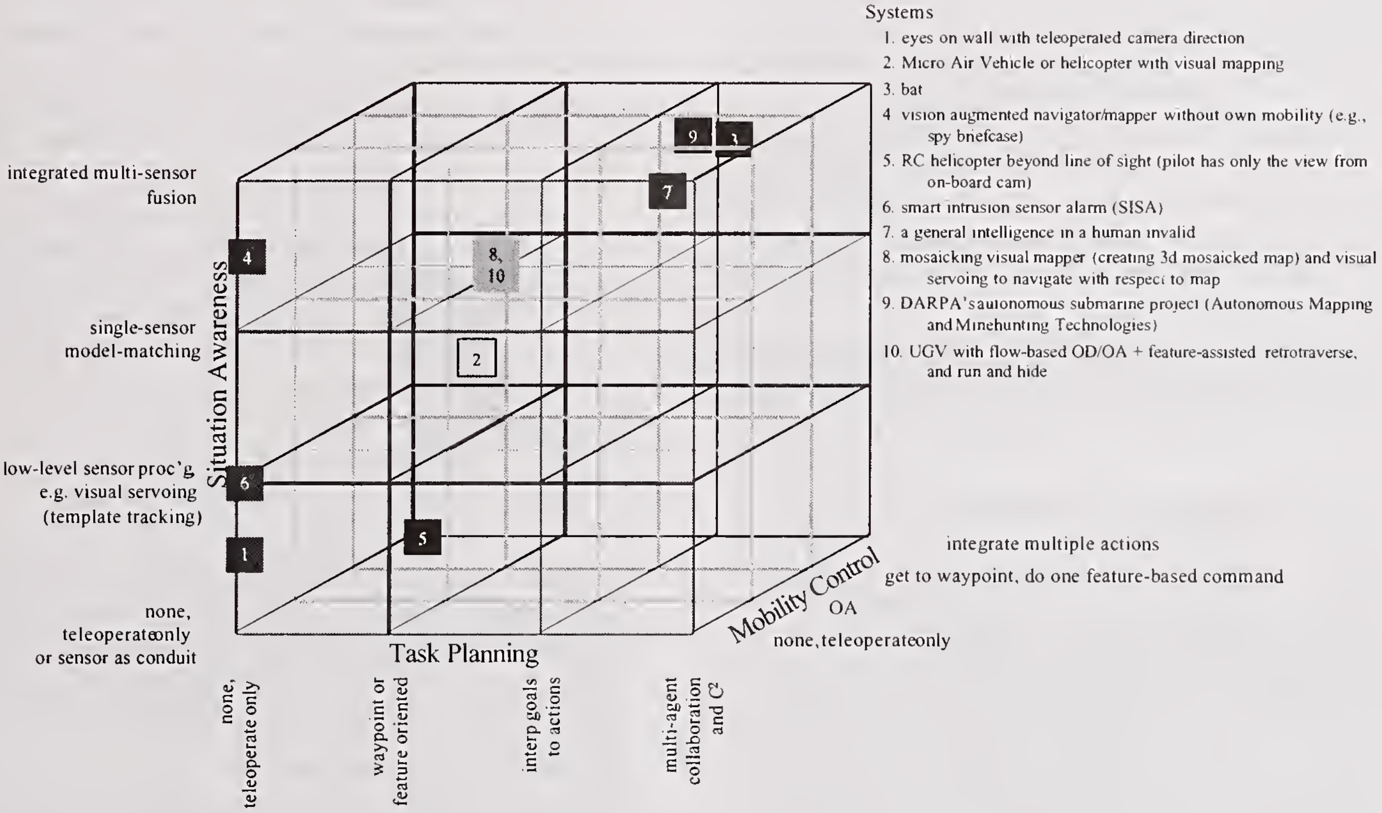

Systems
1. eyes on wall with teleoperated camera direction
2. Micro Air Vehicle or helicopter with visual mapping
3. bat
4. vision augmented navigator/mapper without own mobility (e.g., spy briefcase)
5. RC helicopter beyond line of sight (pilot has only the view from on-board cam)
6. smart intrusion sensor alarm (SISA)
7. a general intelligence in a human invalid
8. mosaicking visual mapper (creating 3d mosaicked map) and visual servoing to navigate with respect to map
9. DARPA's autonomous submarine project (Autonomous Mapping and Minehunting Technologies)
10. UGV with flow-based OD/OA + feature-assisted retrotraverse, and run and hide
Situation Awareness
integrated multi-sensor fusion
single-sensor model-matching
low-level sensor proc'g, e.g. visual servoing (template tracking)
none, teleoperateonly or sensor as conduit
Task Planning
none, teleoperate only
waypoint or feature oriented
interp goals to actions
multi-agent collaboration and C²
Mobility Control
integrate multiple actions
get to waypoint, do one feature-based command
OA
none, teleoperateonly
9
3
7
4
8, 10
2
6
5
1

Figure 2 -- Three-Dimensional Intelligence Space

3 Graphical Parameter Space

A three-dimensional graphical approach was used to illustrate where various systems and system designs fell in the overall parameter space (Figure 2). This shows a particular three axes in the parameter space, recognizing that the whole estimation and metrics space is highly multi-dimensional. Several such charts were prepared, but no canonical axes were identified that best serve all analysis purposes for all autonomous systems. The figure shows axes of situation awareness, mobility, and task planning as creating a 3D intelligence space. A variety of autonomous and non-autonomous systems are included in the figure to highlight key parts of the resulting space.

The representation's key weakness is inherent in the choice of any set of 3 dimensions– key information from a fuller, higher-dimensional space is lost. Also problematic is the apparent linearity between ticks along any axis – what conclusions can be drawn by systems shown N ticks apart? Still, there is the strong sense that this captures something fundamental and accurate about the intelligence present in a variety of compared systems. The primary difficulty with this approach, however, remains the subjective judgement that only a small number of axes is enough to grasp the entire intelligence space.

4 An Intelligence Quotient?

We have been asked at various junctures to provide metrics for autonomous systems development, in a similar vein to those provided by (for instance) engineers working in other disciplines who do not hesitate to propose metrics. That has been a difficult request to answer, until the various exercises reported above led us to a key conclusion: Mature technologies can support more precise performance targets than immature technologies. For instance, a group researching automatic target recognition (ATR) can aim to decrease the false alarm rate by 5%. However, what similar metric applies to the broader aim of "increase intelligent autonomy"?

This section discusses a reservation about characterizing intelligence, then proposes levels of capability that arc our best-yet "intelligence quotient" for autonomous systems.

4.1 A Philosophical Reservation

Answering the above question may depend on how the question is phrased, but consider this goal: enable autonomous dynamic mission replanning, based on discovered targets and conditions expected in the target area, while out of communication with the human operator. Several questions spring to mind. What technologies apply? What are their margins for improvement? Do we even know what is necessary to achieve the goal? One approach is to consider finer-grained technologies rather than the broad term of "autonomy". For instance, the following appear more susceptible to metrification.

- Decrease route planning time-to-plan by 20% given contingencies of type A.
- Increase ATR reliability for particular target/environment pairs by 10%.
- Increase situation recognition capability by increasing contingency representation flexibility by 10 times.

We conclude that "intelligent autonomy" is an immature "technology" that is actually a composition of underlying technologies, all of varying maturities. A small set of examples of component technologies with clear deficiencies (compared to human-level capabilities) follows.

- Sensor data interpretation
- Situation awareness and assessment
- Communication
    - Efficient – perhaps better named "data communication" (bandwidth, rates, etc)
    - Effective – perhaps better named "knowledge communication" (content, concepts, transparency of thought processes)
- Knowledge representation – know, represent and share:
    - What data toward what goals in what timeframes?
    - Why does datum A or set of data B matter?
    - Timeliness of concern
        + Damage is expected to occur by time T (e.g., hostile strike group detected headed for barrier)
        + Unless used by time T, data C not useful (e.g., a moving surface-to-air-missile launcher is dctected 1 mile from bunker moving 10 mph - must use information within 6 minutes)
    - Relatedness of data
- Collaboration
    + Understand others' goals
    + Infer intent from observed behavior

Thus finding ways to divide intelligence and autonomy into appropriate sub-technologies that can be weighed and combined properly is a critical problem facing this effort. Lacking such a reliable analysis tool, we next consider one way to approach its formulation.

4.2 Orders of Intelligence

Given the above reservation, let us proceed to characterize intelligence by asking: how hard is a planning and execution problem? Time to plan (TTP) depends on the size of the planning problem, but Moore's Law will reduce TTP significantly by increasing the feasible size of planning problems. However, TTP also depends on (a) the planning approach (algorithms and representations) and (b) the problem complexity. Size of the problem is the easiest to provide metrics for. The other two factors are used to modulate the metrics. If a planning agent is only concerned with a certain time horizon (e.g., 10 milliseconds, 1 hour, 1 day), the level of detail it considers is similarly bounded. Thus planning problems can be of similar sizes whether at the level of a single vehicle or a fleet of vehicles.

There are numerous planning approaches. For well-characterized and well-formulated domains, search in a pre-defined state space is satisfactory. For other problems, current pure research effots are unable to provide a well-defined solution. More pragmatically, planning and execution systems can use a variety of hybrid approaches, the integration of which pose at least engineering issues.

Problem complexity addresses characteristics beyond the simple size of the problem. The characteristics that make planning, estimation and control difficult include the following elements. Since planning needs to be concerned with what can be expected to occur, it must be concerned with expected results from estimation and control, that are affected by the following elements.

- observability – the degree of hidden state (in controlled system or in situation being monitored)
- complexity of the controllable system. E.g., number and type of actuators, static and dynamic stability of the vehicle.
- situation awareness complexity. E.g.:
  - number of elements
  - interactions between elements
  - dynamism (e.g., likelihood to loose lock in tracking subsystem)
  - model complexity for target identification (e.g., 2D image templates, 3D shape, functional analysis based on shape, behavioral)
  - degree to which situation awareness (SA) fulfills expectations
- number of interacting agents. Especially if multiple agents are simultaneously planning
- number of plan elements in flux simultaneously. E.g., (a) is plan in place before SA is received, or (b) is SA being integrated while plan using it is being created? Regarding example (a) consider the plan "go to area X and find tanks" (where "tanks" will be bound to those found by SA), whereas for (b) consider what the system needs to do when it finds itself unexpectedly under attack from unknown quarters.
- number of levels in planning problem due to (i) number of elements, (ii) number of time horizons, etc.

One approach to creating metrics for these problems is to classify problems from the domain into nominal orders of difficulty, then set targets for various demonstrations which move along the spectrum of difficulty. For instance, reasonable goals might be created by aiming to solve a problem in 1 second in each demo year, where the size and complexity of the problem increases over time. Based on the nominal characterization below of levels of difficulty, the solvable problem size could increase from 10^0 in demo 1 (say year 2), to 10^1 in demo 2 (year 4), and 10^2 in demo 3 (year 6). This folds together the expected advances in processor speed and capacity embodied in Moore's Law with improvements in planning approaches resulting from pure and applied research progress. Table 1 captures this approach and leaves space for additional metrics at various levels of maturity.

	10^0	10^1	10^2	10^3
Demo 1	1 second (TTP0)			
Demo 2		1 second (TTP1)		
Demo 3			1 second (TTP2)	
Beyond				1 second (TTP3)

Table 1 -- Problem Size, and Plan for Increasing Demonstrable Complexity

4.3 Nominal candidate orders of intelligence

The following lists indicate relative order of magnitude capabilities that could be grouped together to assess the maturity of a system's intelligence. These are illustrative, not final. Order 0 activities may exist in preliminary commercial research forms or may need applied research and engineering to be fielded. Higher order activities are believed to be beyond the current state of the art.

Order 0 activities:

- Single vehicle plans including (a) multi-waypoint path planning and execution cognizant of known threats, (b) obstacle avoidance given some warning, (c) deck landing in relatively benign environment
- Multiple vehicle plans, for non-interacting vehicles
- Plan to search area of regard (AOR) for target, where AOR is essentially flat and open, and target can be found by template matching.
- Re-plan communication relay service due to disruption of channel, using prior known assets.
- Re-plan for changed objective, where accomplishment of the objective is in the future from the current time-horizon.
- Re-plan task particulars due to change in SA. E.g., arrive in kill box and discover that the targets to be hit are tanks instead of a column of trucks.
- identify targets of opportunity based on their appearance

Order 1 activities:

- single vehicle obstacle avoidance given less warning and/or more constraints on response (e.g., in confined airspace due to terrain or other vehicles, near vehicle limits for responsiveness)

- single vehicle deck landing in moderate sea state and/or moderate visibility
- Plan to act as autonomous communication relay between moving communication partners, where the partners are moving in ways that are expected to disrupt communication within foreseeable future. Thus plan must include a plan to identify and involve additional communication relays. Alternative contingencies would include planning for disruptions that might occur due to weather, jamming, or other hostile activity.
- Re-plan for changed objective, where accomplishment is within current time-horizon, requiring current SA to be integrated while planning is underway using the being-acquired SA.
- identify targets of opportunity based on their appearance where (e.g.) detection depends on sensor angle, so vehicle must do more extensive search to cover the space of AOR-cross-sensor-attitude. E.g., tanks at edge of forest need to be sensed from the open side. Vehicle should understand the constraints (not just fly more lanes of a survey pattern).
- multi-vehicle plans for interacting vehicles
- strike group flight plan through waypoints and around known threats
- re-plan task goals due to change in SA. E.g., while on wild weasel mission switch to coordinated multi-vehicle SAM attack.

Order 2 activities:

- single vehicle deck landing in high sea state and/or low visibility and/or high and gusty winds
- coordinated obstacle avoidance for a strike group flying very close together

Order 3 activities:

- identify targets of opportunity based on their behavior (from prior planning/SA need model of behavior and identification of behavior based on model)

These characterizations build on those detailed in Section 2 – as vehicles increase their ranking in the Orders of Intelligence, they exhibit more capability in the parameter spaces. For example, consider the multi-vehicle collaboration factor of **degree of coordination / cooperation among the agents**. The Order 0 system includes *multiple vehicle plans for non interacting vehicles*. This could include an system that distributes the team goals among the individual agents for separate completion. The Order 1 system includes a higher level capability in this area of *multi-vehicle plans for interacting vehicles*. Here agents can communicate to one another when they fail or if they are able to take on an increased set of tasks. The Order 2 system increases the requirements on coordination and cooperation to *coordinated obstacle avoidance for a strike group flying very close together*. The system will be required to share situation awareness information and plan coordinated responses at very short time constants.

5 Metrics For Costs

To create a plan for funding toward a goal, an assessment must be made of the state of the technology against the require capabilities. Figure 3 shows such an assessment. It was constructed by asking technology experts to determine the state of maturity of their technologies for solving various parts of a vehicle's parameter space. The colors indicate technological maturity levels:

red	pure research needed (6.1)
yellow	applied research needed (6.2)
green	ready for engineering (6.3)
blank	not applicable

Although this is not a measure of intelligence per se, it supports analyses leading to the construction of intelligence vehicles and groups of vehicles.

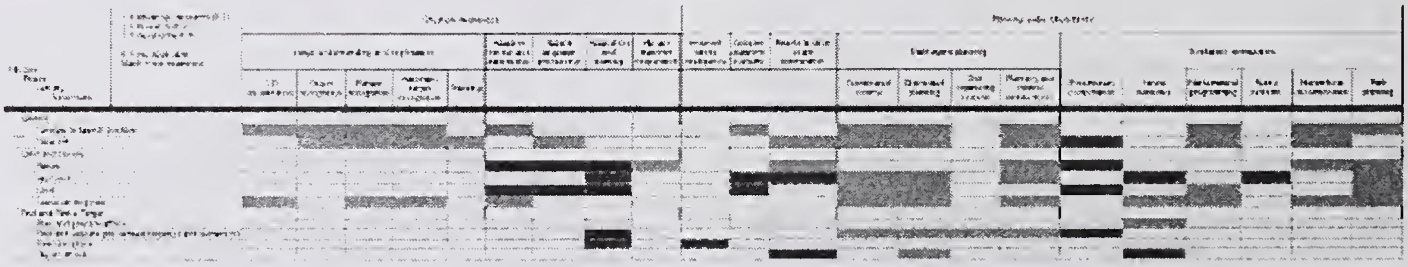


Figure 3 -- Technology Roadmap (partial).
Columns are technologies considered appropriate for addressing the domain, while rows are elements of the vehicle's parameter space.

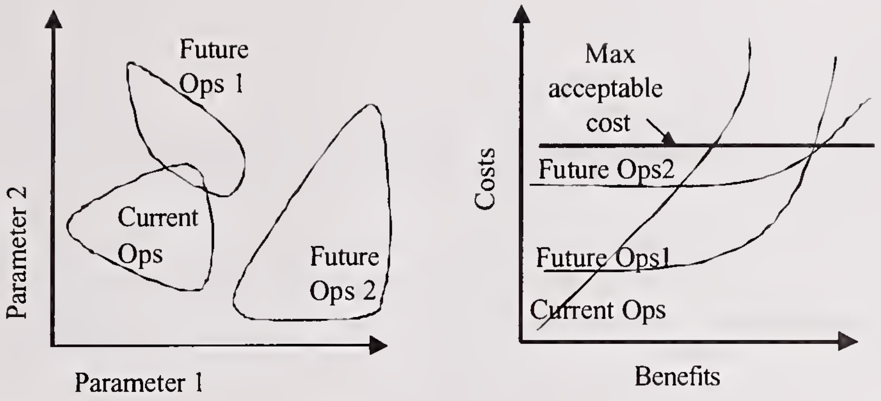


Figure 4 -- Cost/Benefit Analysis.
(left) Where in the parameter space future intelligently autonomous operations exist.
(right) Part of a cost-benefit analysis to select among various operations based on cost.

The notional charts in Figure 4 illustrate how required capabilities can be mapped against mission descriptions of current and future operations, to help determine which are more valued, and to help determine which are expected to be more expensive. Formal methods for such cost projections would be very helpful.

6 Future Directions

Substantial work has been done in applying valuations to multi-attribute (multi-criteria) problems. Besides a number of good textbooks (e.g., [1,2]), various techniques have been formalized to assist in this process. We intend to extend the work reported here by investigating and applying formal tools to the domain characteristics discussed above.

7 Summary

The intelligence of an autonomous vehicle is a complex multi-dimensional characteristic evaluated in a wide variety of dynamic situations, for which no obvious algorithmic measures exist. Several attempts to analyze system complexity and intelligence have been presented in this paper that are drawn from work done for recent and current projects working toward intelligent autonomous vehicles. These analyses have sought to uncover the collaboration, planning and situational awareness challenges facing an autonomous vehicle in difficult conditions, to assist engineers and sponsors in focusing project efforts. Although the analyses reported here have been useful first steps toward the significantly complex vehicles imagined, more work is clearly required before intelligence and intelligent systems can be automatically analyzed and measured.

References

1. Clemen, Robert T., *Making Hard Decisions: An introduction to decision analysis*, PWS-Kent Publishing Co., Boston, 1991

2. French, Simon, *Decision Theory: An introduction to the Mathematics of Rationality*, Ellis Harwood, Ltd. Chichester, West Sussez, England, 1986

3. RoboCup-97: Robot Soccer World Cup I, Springer-Verlag, 1998.

Evolution of Mobile Agents

Timothy K. Shih
Multimedia Information NEtwork (MINE) Lab
Department of Computer Science and Information Engineering
Tamkang University
Tamsui, Taipei Hsien, Taiwan 251, R.O.C.
email: TSHIH@CS.TKU.EDU.TW

Abstract

Mobile agents are powerful. A mobile agent can travel on the Internet, perform tasks, and report to its owner the achievement. Mobile agent techniques are used in E-commerce, distributed applications, distance learning, and others. However, it is hard to find a strategic method, which tells how mobile agents should behave on the Internet. In this paper, we propose such a mechanism. Based one the concepts of Food Web, one of the laws that we may learn from the natural besides neural networks and genetic algorithms, we propose a theoretical computation model for mobile agent evolution on the Internet. We define an agent niche overlap graph and agent evolution states. We also propose a set of algorithms, which is used in our multimedia search programs, to simulate agent evolution. Agents are cloned to live on a remote host station based on three different strategies: the brute force strategy, the semi-brute force strategy, and the selective strategy. Evaluations of different strategies are discussed. Guidelines of writing mobile agent programs are proposed. The technique can be used in distributed information retrieval which allows the computation load to be added to servers, but significantly reduces the traffic of network communication.

1 Introduction

Mobile agents are software programs that can travel over the Internet. Mobile search agents find the information specified by its original query user on a specific station, and send back search results to the user. Only queries and results are transmitted over the Internet. Thus, unnecessary transmission is avoided. In other words, mobile agent computing distributes computation loads among networked stations and reduces network traffic.

The environment where mobile agents live is the Internet. Agents are distributed automatically or semi-automatically via some communication paths. Therefore, agents meet each other on the Internet. Agents have the same goal can share information and cooperate. However, if the system resource (e.g., network bandwidth or disk storage of a station) is insufficient, agents compete with each other. These phenomena are similar to those in the ecosystem of the real world. A creature is born with a goal to live and reproduce. To defense their natural enemies, creatures of the same species cooperate. However, in a perturbation in ecosystems, creatures compete with or even kill each other. The natural world has built a law of balance. Food web (or food chain) embeds the law of creature evolution. With the growing popularity of Internet where mobile agents live, it is our goal to learn from the natural to propose an agent evolution computing model over the Internet. The model, even it is applied only in the mobile agent evolution discussed in this paper, can be generalized to solve other computer science problems. For instance, the search problems in distributed Artificial Intelligence, network traffic control, or any computation that involves a large amount of concurrent/distributed computation. In general, an application of our Food Web evolution model should have the following properties:

- The application must contain a number of concurrent events.
- Events can be simulated by some processes, which can be partitioned into a number of groups according to the properties of events.
- There must exists some consumer-producer relationships among groups so that dependencies can be determined.
- The number of processes must be large enough.

For instance, with the growing popularity of Internet, Web-based documentation are retrieved via some search engine. Search processes can be conducted as several concurrent events distributed among Internet stations. These search events of the same kind (e.g., pursuing the same document) can be formed in a group. Within these agent groups, search agents can provide information to each other. Considering the amount of Web sites in the future, the quantity of concurrent search events is reasonably large.

We have surveyed articles in the area of mobile agents, personal agents, and intelligent agents. The related works are discussed in section 2. Some terminologies and definitions are given in section 3, where we also introduce the detail concepts of agent communication network. In our model, an agent evolves based on state transitions, which are also discussed. A graph theoretical model describes agent dependencies and competitions is also given. Agent evolution computing algorithms are addressed in section 4. And finally, we discuss our conclusions in section 5.

2 Related Works

The concept of mobile agent is discussed in several articles [3, 4]. Agent Tcl, a mobile-agent system providing navigation and communication services, security mechanisms, and debugging and tracking tools, is proposed in [1]. The system allows agent programs move transparently between computers. A software technology called Telescript, with safety and security features, is discussed in [7]. The mobile agent architecture, MAGNA, and its platform are presented in [3]. Another agent infrastructure is implemented to support mobile agents [4]. A mobile agent technique to achieve load balancing in telecommunications networks is proposed in [6]. The mobile agent programs discussed can travel among network nodes to suggest routes for better communications. Mobile service agent techniques and the corresponding architectural principles as well as requirements of a distributed agent environment are discussed in [2].

3 Definitions

Agents communicate with each other since they can help each other. For instance, agents share the same search query should be able to pass query results to each other so that redundant computation can be avoided. An *Agent Communication Network* (ACN) serves this purpose. Each node in an ACN represents an agent on a computer network node, and each link represents a logical computer network connection (or an agent communication link). Since agents of the same goal want to pass results to each other, agent communication relations can be described in a complete graph. Therefore, an ACN of agents hold different goals is a graph of complete graphs. Since agents can have multiple goals (e.g., searching based on multiple criteria), an agent may belong to different complete graphs.

We define some terminologies used in this paper. A *host station* (or *station*) is a networked workstation on which agents live. A *query station* is a station where a user releases a query for achieving a set of goals. A station can hold multiple agents. Similarly, an agent can pursue multiple goals. An *agent society* (or *society*) is a set of agents fully connected by a complete graph, with a common goal associated with each agent in the society. A goal belongs to different agents may have different priorities. An agent society with a common goal of the same priority is called a *species*. Since an agent may have multiple goals, it is possible that two or more societies (or species) have intersections. A *communication cut set* is a set of agents belong to two distinct agent societies, which share common agents. The removing of all elements of a communication cut set results in the separation of the two distinct societies. An agent in a communication cut set is called an *articulation agent*. Since agent societies (or species) are represented by complete graphs and these graphs have communication cut sets as intersections, articulation agents can be used to suggest a shortest network path between a query station and the station where an agent finds its goal. Another point is that an articulation agent can hold a *repository*, which contains the network communication statuses of links of an agent society. Therefore, network resource can be evaluated when an agent checks its surviving environment to decide its evolution policy.

An agent evolves. It can react to an environment, respond to another agent, and communicate with other agents. The evolution process of an agent involves some internal states. An agent is in one of the following states after it is born and before it is killed or dies of natural:

- **Searching**: the agent is searching for a goal
- **Suspending**: the agent is waiting for enough resource in its environment in order to search for its goal
- **Dangling**: the agent loses its goal of surviving, it is waiting for a new goal
- **Mutating**: the agent is changed to a new species with a new goal and a possible new host station

An agent is born to a *searching state* to search for its goal (i.e., information of some kind). All creatures must have goals (e.g., search for food). However, if its surviving environment (i.e., a host station) contains no enough resource, the agent may transfer to a *suspending state* (i.e., hibernation of a creature). The searching process will be resumed when the environment has better resources. But, if the environment is lack of resources badly (i.e., natural disasters occur), the agent might be killed. When an agent finds its goal, the agent will pass the search results to other agents of the same kind (or same society). Other agents will abort their search (since the goal is achieved) and transfer to a *dangling state*. An agent in a dangling state can not survive for a long time. It will die after some days (i.e., a duration of time). Or, it will be re-assigned to a new goal with a possible new host station, which is a new destination where the agent should travel. In this case, the agent is in a *mutating state* and is reborn to search for the new goal. Agent evolution states keep the status of an agent. In order to maintain the activity of agents, in a distributed computing environment, we use message passing as a mechanism to control agent state transitions.

Agents can suspend/resume or even kill each other. We need a general policy to decide which agent is killed. By our definition, a species is a set of agents of the same goal with a same priority. It is the priority of a goal we base on to discriminate two or more species.

We need to construct a direct graph which represents the dependency between species. We call this digraph an *species food web* (or *food web*). Each node in the graph represents a species. All species of a connected food web (i.e., a graph component of the food web) are of the same goal with possibly different priorities. We assume that, different users at different host stations may issue the same query with different priority. Each directed edge in the food web has an origin represents a species of a higher goal priority and has a terminus with a lower priority. Since an agent (and thus a species) can have multiple goals which could be similar to other agents, each goal of an articulation agent should have an associated food web. Therefore, the food web is used as a competition base of agents of the same goal in the same station.

Each food web describes goal priority dependencies of species. Form a food web, we can further derive an *niche overlap graph*. In an ecosystem, two or more species have an *ecological niche overlap* (or *niche overlap*) if and only if they are competing for the same resource. A *niche overlap graph* can be used to represent the competition among species. The niche overlap graph is used in our algorithm to decide agent evolution policy and to estimate the effect when certain factors are changed in an agent communication network. Based on the niche overlap graph, the algorithm is able to suggest strategies to re-arrange policies so that agents can achieve their highest performance efficiency. This concept is similar to the natural process that recover from perturbations in ecosystems.

4 Agent Evolution Computing

The algorithms proposed in this section use the agent evolution states and the niche overlap graphs discussed for agent evolution computing. An agent wants to search for its goal. At the same time, since the searching process is distributed, an agent wants to find a destination station to clone itself. Searching and cloning are essentially exist as a *co-routing relation*. A co-routine can be a pair of processes. While one process serves as a producer, another serves as a consumer. When the consumer uses out of the resource, the consumer is suspended. After that, the producer is activated and produces the resource until it reaches an upper limit. The producer is suspended and the consumer is resumed. In the computation model, the searching process can be a consumer, which need new destinations to proceed search. On the other hand, the cloning process is a producer who provides new URLs.

Agent evolution on the agent communication network is an asynchronous computation. Agents live on different (or the same) stations communicate and work with each other via agent messages. The searching and the cloning processes of an agent may run as a co-routine on a station. However, different agents are run on the same or separated stations concurrently. We use a formal specification approach to describe the logic of our evolution computation. Formal specifications use first order logic, which is precise. In this paper, we use the Z specification language to describe the model and algorithms.

Each algorithm or global variable in our discussion has two parts. The expressions above a horizontal line are the signatures of predicates, functions, or the data types of variables. Predicates and functions are constructed using quantifiers, logic operators, and other predicates (or functions). The signature of a predicate also indicate the type of its formal parameters. For instance, $Agent \times Goal \times Host\_Station$ are the types of formal parameters of predicate $Agent\_Search$. The body, as the second part of the predicate, is specified below the horizontal line.

We use some global variables through the formal specification. The variable $goal\_achieved$ is set to $TRUE$ when the search goal is achieved, $FALSE$ otherwise. We also use two watermark variables, α and β, where α is the basic system resource requirement and β is the minimal requirement. Note that, α must be greater than β so that different levels of treatment are used when the resource is not sufficient.

Global Variables and Constants

$$\begin{array}{|l} goal\_achieved : Goal\_Achieved \\ \alpha : REAL \\ \beta : REAL \\ \hline \alpha > \beta \end{array}$$

Algorithm $Agent\_Search$ is the starting point of agent evolution simulation. If system resource meets a basic requirement (i.e., α), the algorithm activates an agent in the searching state within a local station. If the search process finds its goal (e.g., the requested information is found), the goal is achieved. Goal abortion of all agents in a society results in a dangling state of all agents in the same society (including the agent who finds the goal). At the same time, the search result is sent back to the original query station via *Query_Return_URL*. Suppose that the goal can not be achieved in an individual station. the agent is cloned in another station (agent propagation). The *Agent_Clone* algorithm is then used. On the other hand, the agent may be suspended or even killed if the system resource is below the basic requirement (i.e., $Resource\_Available(A, G, X) < \alpha$). In this case, algorithms *Agent_Suspend* is used if the resource available is still feasible for a future resuming of the agent. Otherwise, if the resource is below the minimal requirement, algorithm *Agent_Kill* is used.

Agent Searching Algorithm

```
$Agent\_Search : Agent \times Goal \times Host\_Station$

$\forall A : Agent, G : Goal, X : Host\_Station \bullet$
    $Agent\_Search(A, G, X) \Leftrightarrow$
        $Resource\_Available(A, G, X) \geq \alpha \Rightarrow$
            $[G \in Local\_Search(A, X) \Rightarrow$
                $Abort\_All(A \upharpoonright Agent\_Society) \wedge$
                $send\_result(X.URL,$
                    $G.Query\_Return\_URL) \wedge$
                $goal\_achieved = TRUE$
            $\vee\ G \notin Local\_Search(A, X) \Rightarrow$
                $Agent\_Clone(A, G,$
                    $A \upharpoonright Agent\_Society)]$
        $\vee\ Resource\_Available(A, G, X) \geq \beta \Rightarrow$
            $Agent\_Suspend(A, G, X)$
        $\vee\ Resource\_Available(A, G, X) < \beta \Rightarrow$
            $Agent\_Kill(A, G, X)$
```

Agent cloning is achieved by the *Agent_Clone* algorithm. When the cloning process wants to find new stations to broadcast an agent, two implementations can be considered. The first is to collect all URLs of stations found by one search engine. But, considering the network resource available. the implementation may check for the common URLs found by two or more search engines. New URLs are collected by the *Search_For_Stations* algorithm, which is invoked in the agent cloning algorithm. Agent propagation strategy decides the computation efficiency of our model. In this research, we propose three strategies:

- the brute force agent distribution
- the semi-brute force agent distribution, and
- the selective agent distribution.

The first strategy simply clone an agent on a remote station, if the potential station contains information that helps the agent to achieve its goal. The semi-brute force strategy. however, finds another agent on a potential station, and assigns the goal to that agent. The selective approach not only try to find a useful agent, but also check for the goals of that agent. Cloning strategies affect the size of agent societies thus the efficiency of computation.

Agent Cloning Algorithm: the Brute Force Strategy

```
$Agent\_Clone : Agent \times Goal \times Agent\_Society$

$\forall A : Agent, G : Goal, S : Agent\_Society \bullet$
    $Agent\_Clone(A, G, S) \Leftrightarrow$
        $[\forall X : Host\_Station \bullet$
                $X \in Search\_For\_Stations(G) \Rightarrow$
            $(\exists A' : Agent \bullet A' = copy(A) \wedge$
            $X.Agent\_Set = X.Agent\_Set \cup \{ A' \} \wedge$
            $S = S \cup \{ A' \} \wedge$
            $Agent\_Search(A', G. X))]$
        $\vee\ [Search\_For\_Stations(G) = \emptyset \Rightarrow$
            $goal\_achieved = FALSE]$
```

The brute force agent distribution strategy makes a copy of agent A, using the *copy* function. in all stations returned by the *Search_For_Stations* algorithm. Agent set in each station is updated and the society S where agent A belongs is changed. Agent A', a clone of agent A is transmitted to station X for execution.

Agent Cloning Algorithm: the Semi-brute Force Strategy

```
$Agent\_Clone : Agent \times Goal \times Agent\_Society$

$\forall A : Agent, G : Goal, S : Agent\_Society \bullet$
    $Agent\_Clone(A, G, S) \Leftrightarrow$
        $[\forall X : Host\_Station \bullet$
            $X \in Search\_For\_Stations(G) \Rightarrow$
            $[\exists A' : Agent \bullet A' \in X.Agent\_Set \Rightarrow$
                $(A'.Goal\_Set = A'.Goal\_Set \cup$
                        $\{ G \} \wedge$
                    $S = S \cup \{ A' \} \wedge$
                    $Agent\_Search(A', G, X))]]$
        $\vee\ [Search\_For\_Stations(G) = \emptyset \Rightarrow$
            $goal\_achieved = FALSE]$
```

The semi-brute force agent distribution approach is similar to the brute force approach, except that it does not make a copy of the agent but give the goal to an agent on its destination station. The agent which accepts this new goal (i.e., A') is activated for the new goal in its belonging station.

Agent Cloning Algorithm: the Selective Strategy

```
Agcnt_Clonc : Agcnt × Goal × Agcnt_Socicty
──────────────
∀ A : Agent, G : Goal, S : Agent_Society •
    Agent_Clone(A, G, S) ⇔
        [∀ X : Host_Station •
            X ∈ Search_For_Stations(G) ⇒
            [∃ A′ : Agent • A′ ∈ X.Agent_Set ⇒
                [G ∈ A′.Goal_Set ⇒
                    S = S ∪ A′ ↾ Agent_Society
                ∨ G ∉ A′.Goal_Set ⇒
                    (A′.Goal_Set =
                        A′.Goal_Set ∪ { G }
                    ∧ S = S ∪ { A′ })]
                ∧ Agcnt_Scarch(A′, G, X)]
            ∨ [X.Agent_Set = ∅ ⇒
                [∃ A″ : Agent • A″ = copy(A) ∧
                X.Agent_Set = { A″ } ∧
                S = S ∪ { A″ } ∧
                Agent_Search(A″, G, X)]]]
        ∨ [Search_For_Stations(G) = ∅
            ⇒ goal_achieved = FALSE]
```

The last approach is more complicate. The selective approach of cloning algorithm must check whether there is another agent in the destination station (i.e., X). If so, the algorithm checks whether the agent (i.e., A') at that station shares the same goal with the agent to be cloned. If two agents share the same goal, there is no nccd of cloning anothcr copy of agcnt. Basically, thc goal can be computed by the agent at the destination station. In this case, the union of the two societies is necessary (i.e., $S = S \cup A' \upharpoonright Agent\_Society$). On the other hand, if the two agents do not have a common goal, to save computation resource, we may ask the agent at the destination station to help searching for an additional goal. This case makes a re-organization of the society where the source agent belongs. The result also ensure that the number of agents on the ACN is kept in a minimum. Whether the two agents share the same goal, the *Agent_Search* algorithm is used to search for the goal again. In this case, Agent A' is physically transmitted to station X for execution. When there is no agent running on the destination station, we need to increase the number of agents on the ACN by duplicating an agent on the destination station (i.e., the invocation of $A'' = copy(A)$). The society is reorganized. And the *Agent_Search* algorithm is called again. In the acse that no new station is found by thc *Scarch_For_Stations* algorithm, thc goal is not achieved.

The agent search and agent clone algorithms use somc auxiliary algorithms, which arc discussed as follows. The justification of system resource available depends on agent policy, as defined in *A.Policy*. Agent policy is a set of factors indicated by name tags (e.g., *NETWORK_BOUND*). The estimation of resources is represented as a real number, which is computed based on *X.Resource* of station X. Note that, in the algorithm, $w1$ and $w2$ are weights of factors ($w1 + w2 = 1.0$). Wc only dcscribcs somc cascs of using agcnt policies. Other cases are possible but omitted. Moreover, we consider the priority of goal G. If the priority is lower than some watermark (i.e., $G.Priority < \theta$), we let $r1$ be a constant less than 1.0. Therefore, resources are reserved for other agents. On the other hand, if the priority is high, we consider the value returned by *Rcsourcc_Available* should be high. Thus the potential agent can proceed its computation immediately. The values of θ and ω depend on agent applications.

Auxiliary Algorithms

```
Resource_Available : Agent × Goal × Host_Station →
    REAL
──────────────
∀ A : Agent, G : Goal, X : Host_Station, R : REAL •
    ∃ w1, w2, r1, r2 : REAL •
    Rcsourcc_Availablc(A, G, X) = R ⇔
        [NETWORK_BOUND ∈ A.Policy ⇒
            R = X.Resource.Network
        ∨ CPU_BOUND ∈ A.Policy ⇒
            R = X.Resource.CPU
        ∨ MEMORY_BOUND ∈ A.Policy ⇒
            R = X.Resource.Memory
        ∨ CPU_BOUND ∈ A.Policy ∧
        MEMORY_BOUND ∈ A.Policy ⇒
            R = X.Resource.CPU * w1+
                X.Resource.Memory * w2 ∧
            w1 + w2 = 1.0
        ∨ ...]
        ∧ ∃θ, ω : Priority •
            [G.Priority < θ ⇒
                (R = R * r1 ∧ r1 < 1.0)
            ∨ G.Priority > ω ⇒
                (R = R * r2 ∧ r2 > 1.0)]
```

The above algorithms describe how an agent evolves from a state to another. How agents affect each other dcpends on thc systcm rcsourcc availablc. Howcvcr, in an ACN, it is possible that agents suspend or even kill each other, as we described in previous sections. The niche overlap graphs of each goal play an important role. We use the *Agent_Suspend* and *Agent_Kill* algorithms to take the niche overlap graphs of a goal (i.e., *niche_compete*(G)) into consideration. In the *Agent_Suspend* algorithm, if there exists a goal that has a lower priority comparing to the goal of the searching agent, a suspend message is sent to the goal to delay its search (i.e., via *suspend*($G' \upharpoonright Agent$)). The searching agent may be resumed after that since system resources may be released from those goal suspension. In the *Agent_Kill* algorithm, however, a kill message is sent instead (i.e., via *terminate*($G' \upharpoonright Agent$)). The system resource is checked against the minimum requirement β. If resuming is feasible, the *Agent_Search* algorithm in invoked. Otherwise, the system should terminate the searching agent.

```
$Agent\_Suspend : Agent \times Goal \times Host\_Station$
$\forall A : Agent, G : Goal, X : Host\_Station \bullet$
    $Agent\_Suspend(A, G, X) \Leftrightarrow$
        $\exists GS : Goal\_Set \bullet$
            $GS = niche\_compete(G)$
            $\wedge (\forall G' : Goal \bullet G' \in GS \wedge$
                $G'.Priority < G.Priority \Rightarrow$
                    $suspend(G' \uparrow Agent))$
            $\wedge (Resource\_Available(A, G, X) \geq \beta \Rightarrow$
                $Agent\_Search(A, G, X)$
            $\vee Resource\_Available(A, G, X) < \beta \Rightarrow$
                $suspend(A))$
```

```
$Agent\_Kill : Agent \times Goal \times Host\_Station$
$\forall A : Agent, G : Goal, X : Host\_Station \bullet$
    $Agent\_Kill(A, G, X) \Leftrightarrow$
        $\exists GS : Goal\_Set \bullet$
            $GS = niche\_compete(G)$
            $\wedge (\forall G' : Goal \bullet G' \in GS \wedge$
                $G'.Priority < G.Priority \Rightarrow$
                    $terminate(G' \uparrow Agent))$
            $\wedge (Resource\_Available(A, G, X) \geq \beta \Rightarrow$
                $Agent\_Search(A, G, X)$
            $\vee Resource\_Available(A, G, X) < \beta \Rightarrow$
                $terminate(A))$
```

The other auxiliary algorithms are relatively less complicated. Function *Local_Search* takes as input an agent and a station. It returns a set of goals found by the agent in that station. A *match* predicate is used. This match predicate is application dependent. It could be a search program which locates a key word in a Web page, or a request of information from a user (e.g., a survey questionnaire). The *Abort_All* predicate takes as input an agent society and terminates all agents within that society. The *Search_For_Stations* function takes as input a goal and returns a set of host stations. The stations should be selected depending on the *candidate_station* function, which estimates the possibility of goal achievement in a station. This function can be implemented as a Web search engine which looks for candidate URLs. We have omitted some detailed definitions of the above auxiliary algorithms, as well as some primitive functions which are self-explanatory.

```
$Local\_Search : Agent \times Host\_Station \rightarrow Goal\_Set$
$\forall A : Agent, X : Host\_Station, GS : Goal\_Set \bullet$
    $Local\_Search(A, X) = GS \Leftrightarrow$
        $GS = \{ G : Goal \mid G \in A.Goal\_Set \wedge$
            $match(G.Query,$
                $X.Resource.Information) \}$
```

```
$Abort\_All : Agent\_Society$
$\forall S : Agent\_Society \bullet$
    $Abort\_All(S) \Leftrightarrow$
        $\forall A : Agent \bullet A \in S \Rightarrow terminate(A)$
```

```
$Search\_For\_Stations : Goal \rightarrow \mathbb{P}\ Host\_Station$
$\forall G : Goal, X\_Set : \mathbb{P}\ Host\_Station \bullet$
    $Search\_For\_Stations(G) = X\_Set \Leftrightarrow$
        $X\_Set = \{ X : Host\_Station \mid$
            $candidate\_station(G, X) \}$
```

5 Conclusions

Mobile agent based software engineering is interesting. However, in the literature, we did not find any other similar theoretical approach to model what mobile agents should act on the Internet, especially how mobile agents can cooperate and compete. A theoretical computation model for agent evolution was proposed in this paper. Algorithms for the realization of our model were also given.

References

[1] David Kotz, Robert Gray, Saurab Nog, Daniela Rus, Sumit Chawla, and George Cybenko, "Agent Tcl: targeting the needs of mobile computers," IEEE Internet Computing, Vol. 1, No. 4, July 1997, pp. 58 – 67.

[2] S. Krause and T. Magedanz, "Mobile service agents enabling intelligence on demand in telecommunications," in Proceedings of the 1996 IEEE Global Telecommunications Conference, London, UK, 1996, pp 78 – 84.

[3] Sven Krause, Flavio Morais de Assis Silva, and Thomas Magedanz, "MAGNA - a DPE-based platform for mobile agents in electronic service markets," in Proceedings of the 1997 3rd International Symposium on Autonomous Decentralized Systems (ISADS'97), Berlin, Germany, 1997, pp. 93 – 102.

[4] Anselm Lingnau and Oswald Drobnik, "Making mobile agents communicate: a flexible approach," in Proceedings of the 1996 1st Annual Conference on Emerging Technologies and Applications in Communications, Portland, OR, USA, 1996, pp. 180 – 183.

[5] Michael Pazzani and Daniel Billsus, "Learning and Revising User Profiles: The Identification of Interesting Web Sites", Machine Learning, Vol. 27, 1997, pp. 313 – 331.

[6] Ruud Schoonderwoerd, Owen Holland, and Janet Bruten, "Ant-like agents for load balancing in telecommunications networks," in Proceedings of the 1997 1st International Conference on Autonomous Agents, Marina del Rey, California, U.S.A., 1997, pp. 209 – 216.

[7] Joseph Tardo and Luis Valente, "Mobile agent security and telescript," in Proceedings of the 1996 41st IEEE Computer Society International Conference (COMPCON'96), Santa Clara, CA, USA, 1996, pp. 58 – 63.

PART II
RESEARCH PAPERS

5. MODELING THE BIOLOGICAL PROTOTYPES

SOME CONSTRAINTS ON INTELLIGENT SYSTEMS

Autonomous Computation in a Changing World

Stephen Grossberg

Department of Cognitive and Neural Systems, Boston University, 677 Beacon Street, Boston, MA 02215

steve@bu.edu, http://www.cns.bu.edu/Profiles/Grossberg

Among the many possible topics concerning how autonomous intelligent systems should be designed, I will focus on one that is close to work with which I am familiar. A core problem on which much more work needs to be done is how to design systems that can autonomously learn, recognize, and perform complex tasks in a rapidly changing environment. Such self-organizing systems should also be able to interact effectively with humans and other self-organizing systems in order to achieve goals cooperatively.

In order to make the interface between human and system as seamless as possible, biological designs, notably designed inspired by and even emulating brain architectures, will be helpful. The list of possible applications is incredibly long, ranging from autonomous search and data base management tools on the world wide web, medical data base prediction to help doctors and other health professionals, classifiers of complex imagery of multiple types, new approaches to speech perception in noisy multi-speaker environments, and controllers of autonomous mobile robots, to models of normal brain and behavior, and predictions about how different brain lesions can generate the behavioral symptoms of mental disorders.

Available results have already suggested that the brain designs for sensory and cognitive processes differ from, and are even computationally complementary to, the designs for spatial navigation and action. This complementarity can be noticed by observing that cognitive knowledge needs to accumulate in a stable way over a period of years, with new knowledge not accidentally erasing previously learned, but still useful, knowledge. This is the familiar problem of "catastrophic forgetting". In contrast, the parameters that control action need to be continually updated in order to adapt to changes, including damage, to motor effectors. Here, catastrophic forgetting is a useful property. Thus, these systems will need to incorporate new ideas about parallel processing between information subsystems that compute complementary properties.

The design of increasingly autonomous intelligent agents will also require an end-to-end approach, in which all the aspects of perception, cognition, emotion, and action are realized in a single system. Feedback cycles of information processing need to be designed from perception through action and then back to perception again, mediated by feedback through the environment. Such cycles of information processing can evaluate the effects of system performance on the environment, and modify the system where needed to achieve better environmental control. It has also become clear that, in addition to these externally mediated cycles of information processing with the environment, internally mediated feedback is needed to achieve autonomous system properties. Such internal feedback realizes properties of intentionality and attention that are characteristic of biological intelligence. The design of self-organizing feedback systems will require a deeper analysis of nonlinear systems, since various types of nonlinearity are needed to achieve key system properties that depend on feedback, such as the stability of fast learning in a changing environment.

One example of such an autonomous system is the primate cerebral cortex. All sensory and cognitive neocortex is organized into laminar circuits, wherein bottom-up, top-down, and horizontal connections are synthesized into a unified design. Recent modeling (Grossberg, 1999; Grossberg, Mingolla, and Ross, 1997; Grossberg and Raizada, 2000; Grossberg and Williamson, 2000) has clarified how these laminar circuits are designed (Figure 1) to simultaneously achieve at least three properties: (1) stable development and learning of circuit connections and adaptive weights in response to a changing world, thereby providing a solution of the *stability-*

plasticity dilemma; (2) a seamless fusion of bottom-up data-driven processing and top-down intentional processing whereby high-level constraints can selectively focus attention upon important information; and (3) the coherent grouping or binding of spatially distributed information into representations of objects and events, while suppressing noise and weaker groupings, without a loss of analog sensitivity to input values, the so-called property of *analog coherence*.

The design of more subtle decision making processes in an autonomous agent will require more sophisticated cognitive-emotional interactions, whereby the information acquired through cognitive processing is evaluated and selected in terms of internal system values and goals. Such interactions help to direct attention selectively to those subsets of information that predict future success in achieving system goals. Designs for such systems need to be able to use unsupervised learning when no evaluative feedback is available, but to be able to switch to supervised learning whenever such feedback is available. In a self-organizing autonomous learning system, both unsupervised and supervised learning need to be able to operate without a change of system design. Taken together, these constraints point to the development of new types of self-organizing parallel processing systems wherein nonlinear feedback within the system and between system and world help the system to rapidly adapt to a changing world, and thereby to better represent, predict and control it.

Illustrative References

Brown, J., Bullock, D. and Grossberg, S. (1999). How the basal ganglia use parallel excitatory and inhibitory learning pathways to selectively respond to unexpected rewarding cues. *Journal of Neuroscience*, **19**, 10502-10511.

Fiala, J., Grossberg, S. and Bullock, D. (1996). Metabotropic glutamate receptor activation in cerebellar Purkinje cells as substrate for adaptive timing of the classically conditioned eye-blink response. *Journal of Neuroscience*, **16**, 3760-3774.

Grossberg, S. (1999). How does the cerebral cortex work? Learning, attention, and grouping within the laminar circuits of visual cortex. *Spatial Vision*, **12**, 163-187.

Grossberg, S. (1999). The link between brain learning, attention, and consciousness. *Consciousness and Cognition*, **8**, 1-44.

Grossberg, S. (2000). The complementary brain: Unifying brain dynamics and modularity. *Trends in Cognitive Sciences*, **4**, 233-246.

Grossberg, S (2000). The imbalanced brain: From normal behavior to schizophrenia. *Biological Psychiatry*, in press.

Grossberg, S., Boardman, I. and Cohen, M.A. (1997). Neural dynamics of variable-rate speech categorization. Journal of Experimental Psychology: *Human Perception and Performance*, **23**, 481-503.

Grossberg, S. and Merrill, J.W.L. (1996). The hippocampus and cerebellum in adaptively timed learning, recognition, and movement. *Journal of Cognitive Neuroscience*, **8**, 257-277.

Grossberg, S., Mingolla, E. and Ross, W. (1997). Visual brain and visual perception: how does the cortex do perceptual grouping? *Trends in Neurosciences*, **20**, 106-111.

Grossberg, S. and Myers, C.W. (2000). The resonant dynamics of speech peiception: Interword integration and duration-dependent backward effects, *Psychological Review*, in press.

Grossberg, S. and Raizada, R.D.S. (2000). Contrast-sensitive perceptual grouping and object-based attention in the laminar circuits of primary visual cortex. *Vision Research*, **40**, 1413-1432.

Grossberg, S. and Williamson, J.R. (2000). A neural model of how horizontal and interlaiminar connections of visual cortex develop into adult circuits that carry out perceptual grouping and learning. *Cerebral Cortex*, in press.

Figure Caption

Figure 1. Some model cell interactions between the lateral geniculate nucleus (LGN) and cortical areas V1 and V2 for perceptual grouping and attention: Excitatory connections are shown with open symbols. Inhibitory interneurons are shown filled-in black. (a): The LGN provides bottom-up activation to layer 4 via two routes. Firstly, it makes a strong connection directly into layer 4. Secondly, LGN axons send collaterals into layer 6, and thereby also activate layer 4 via the 6 → 4 on-center off-surround path. Thus, the combined effect of the bottom-up LGN pathways is to stimulate layer 4 via an on-center off-surround, which provides divisive contrast normalization of layer 4 cell responses. (b): *Folded feedback* carries attentional signals from higher cortex into layer 4 of V1, via the modulatory 6 → 4 path. Corticocortical feedback axons tend preferentially to originate in layer 6 of the higher area and to terminate in the lower cortex's layer 1, where they can excite the apical dendrites of layer 5 pyramidal cells whose axons send collaterals into layer 6. Several other routes through which feedback can pass into V1 layer 6 exist. Having arrived in layer 6, the feedback is then "folded" back up into the feedforward stream by passing through the 6 → 4 on-center off-surround path. (c): Connecting the 6 → 4 on-center off-surround to the layer 2/3 grouping circuit: like-oriented layer 4 simple cells with opposite contrast polarities compete (not shown) before generating half-wave rectified outputs that converge onto layer 2/3 complex cells in the column above them. Like attentional signals from higher cortex, groupings which form within layer 2/3 also send activation into the *folded feedback* path, to enhance their own positions in layer 4 beneath them via the 6 → 4 on-center, and to suppress input to other groupings via the 6 → 4 off-surround. There exist direct layer 2/3 → 6 connections in macaque V1, as well as indirect routes via layer 5. (d): Top-down corticogeniculate feedback from V1 layer 6 to LGN also has an on-center off-surround anatomy, similar to the 6 → 4 path. The on-center feedback selectively enhances LGN cells that are consistent with the activation that they cause, and the off-surround contributes to length-sensitive (endstopped) responses that facilitate grouping perpendicular to line ends. (e): The entire V1/V2 circuit: V2 repeats the laminar pattern of V1 circuitry, but at a larger spatial scale. In particular, the horizontal layer 2/3 connections have a longer range in V2, allowing above-threshold perceptual groupings between more widely spaced inducing stimuli to form. V1 layer 2/3 projects up to V2 layers 6 and 4, just as LGN projects to layers 6 an 4 of V1. Higher cortical areas send feedback into V2 which ultimately reaches layer 6, just as V2 feedback acts on layer 6 of V1. Feedback paths from higher cortical areas straight into V1 (not shown) can complement and enhance feedback from V2 into V1.

The Neurodynamics of Intentionality in Animal Brains May Provide a Basis for Constructing Devices that are Capable of Intelligent Behavior

Walter J Freeman

Department of Molecular and Cell Biology
University of California, Berkeley CA 94720-3200 USA

ABSTRACT

Intelligent behavior is characterized by flexible and creative pursuit of endogenously defined goals. It has emerged in humans through the stages of evolution that are manifested in the brains and behaviors of other animals. Intentionality is a key concept by which to link brain dynamics to goal-directed behavior. The archetypal form of intentional behavior is an act of observation through time and space, by which information is sought for the guidance of future action. Sequences of such acts constitute the key desired property of free-roving, semi-autonomous devices capable of exploring remote environments that are inhospitable for humans. Intentionality consists of the neurodynamics by which images are created of future states as goals, of command sequences by which to act in pursuit of goals, of predicted changes in sensory input resulting from intended actions (reafference) by which to evaluate performance, and modification of the device by itself for learning from the consequences of its intended actions. These principles are well known among psychologists and philosophers. What is new is the development of nonlinear mesoscopic brain dynamics, by which using chaos theory to understand and simulate the construction of meaningful patterns of neural activity that implement the perceptual process of observation. The prototypic hardware realization of intelligent behavior is already apparent in certain classes of robots. The chaotic neurodynamics of sensory cortices in pattern recognition is ready for hardware embodiments, which are needed to provide the eyes, noses and ears of devices for survival and autonomous operation in complex and unpredictable environments.

Key Words: *Chaos theory, Intentionality, Mesoscopic Brain dynamics, Perception, Reafference*

1.0 Neurodynamics of intentionality in the behavioral act of observation

1.1 The properties of intentionality

The first step in pursuit of an understanding of intentionality is to ask, what happens in brains during an act of observation? This is not a passive receipt of information from the world. It is a purposive action by which an observer directs the sense organs toward a selected aspect of the world and interprets the resulting barrage of sensory stimuli. The concept of intentionality has been used to describe this process in different contexts, since its first use by Aquinas in 1272 [1]. The three salient characteristics of intentionality as it was developed by him are (a) intent or directedness toward some future state or goal, (b) wholeness, and (c) unity [12]. These three aspects correspond to current use of the term in psychology [with the meaning of purpose], in medicine [with the meaning of mode of healing and integration of the body], and in analytic philosophy [with the meaning of the way in which beliefs and thoughts are connected with ("about") objects and events in the world, also known as the symbol-grounding problem].

Intent comprises the endogenous initiation, construction, and direction of behavior into the world. It emerges from brains. Humans, animals and autonomous robots select their own goals, plan their own tactics, and choose when to begin, modify, and stop sequences of action. Humans at least are subjectively aware of themselves acting, but consciousness is not a necessary property of intention. Unity appears in the combining of input from all sensory modalities into *Gestalts*, in the coordination of all parts of the body, both musculoskeletal and autonomic, into adaptive, flexible, yet focused movements. Subjectively, unity appears in the awareness of self and emotion, but again this is not intrinsic to intention. Wholeness is revealed by the orderly changes in the self and its behavior that constitute the development, maturation and adaptation of the self, within the constraints of its genes or design principles, and its material, social and industrial environments. Subjectively, wholeness is revealed in the remembrance of self through a lifetime of change, although the influences of accumulated and integrated experience on current behavior are not dependent on recollection and recognition. In brief, simulation of intentionality should be directed toward replicating the mechanisms by which goal states are constructed, approached and evaluated, and not toward emulating processes of consciousness, awareness, emotion, etc. in machines.

1.2 The limbic system is the chief organ of intentional behavior

Brain scientists have known for over a century that the necessary and sufficient part of the vertebrate brain to sustain minimal intentional behavior is the ventral forebrain, including those components that comprise the external shell of the phylogenetically oldest part of the forebrain, the paleocortex, and the deeper lying nuclei with which the cortex is connected. These components suffice to support remarkably adept patterns of intentional behavior, in dogs after all the newer parts of the forebrain have been surgically removed [17], and in rats with neocortex chemically inactivated by spreading depression [3]. Intentional behavior is severely altered or absent after major damage to the medial temporal lobe of the basal forebrain, as manifested most widely in Alzheimer's disease.

Phylogenetic evidence comes from observing intentional behavior in salamanders, which have the simplest of the existing vertebrate forebrains [21, 28]. The three main parts are sensory (which, as in small mammals, is predominantly olfactory), motor, and associational (Figure 1). These parts can be judged to comprise the limbic system in all vertebrates, but in the salamander they have virtually none of the "add-ons" found in brains of higher vertebrates, hence the simplicity. The associational part contains the primordial hippocampus with its interconnected septum and amygdaloid nuclei, striatal nuclei, which are identified in higher vertebrates as the locus of the functions of spatial orientation (the "cognitive map") and temporal integration in learning (the organization of long and short term memory). These processes are essential, inasmuch as intentional action takes place into the world, and even the simplest action, such as searching for food or evading predators, requires an animal to know where it is with respect to its world, where its prey or refuge is, and what its spatial and temporal progress is during sequences of attack or escape. The feedback loops that support the flow of neural activity in the neurodynamics of intentionality are schematized in Figure 2.

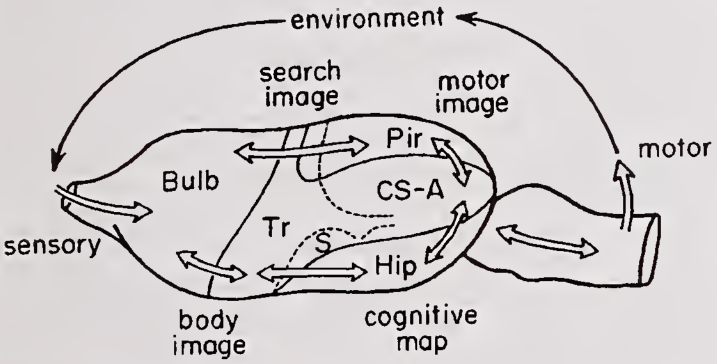


Figure 1. This schematic illustrates the sensory, motor, and associational components of the right hemisphere (seen from above) of the simplest extant vertebrate brain in the salamander. The bidirectional connections between these 3 major subdivisions of the forebrain provide for the macroscopic interactions that support the neurodynamics of the process of intentionality: goal formation, action, perception, and learning from the sensory consequences of the action taken into the environment. These components are form the prototype of the limbic system, which is found in all vertebrate brains, typically buried within exuberant growth of other "add-on" structures that operate in concert with the limbic system.

1.3 Neurodynamic manifestations of intentionality in brain activity of the primary sensory cortices: the EEG (electroencephalogram , 'local field potential')

The crucial question for neuroscientists is, how are the patterns of neural activity that sustain intentional behavior constructed in brains prior to perception? An answer is provided by studies of electrical activity of the primary sensory cortices of animals that trained to respond to conditioned stimuli [2, 8, 10-12, 14, 22, 23]. The construction is not by recall of stored patterns but by pattern formation in distributed nonlinear systems with connections that have been modified cumulatively through learning. The manner in which this take place involves hierarchical ordering of neural activity between microscopic, mesoscopic and macroscopic levels having differing time and space scales. Cortical neurons are selectively activated by sensory receptors and made to generate microscopic activity in the form of trains of action potentials (pulses) on their axons.. These and neighboring neurons by their synaptic interactions form a population forms that "binds" their activity into mesoscopic patterns 14, 18, 19, 29, 30]. These mesoscopic brain activity patterns are revealed by electrical fields of potential (EEGs) generated by interactive masses of neurons are induced by the arrival of stimuli, which trigger sequences of 1st order state transitions. These sequential states in turn converge into integrated macroscopic patterns that occupy the entirety of each cerebral hemisphere and give rise to the global patterns of brain activity, that may be related to the patterns of metabolic activity that are revealed by non-invasive brain imaging (fMRI, PET, SPECT, etc.).

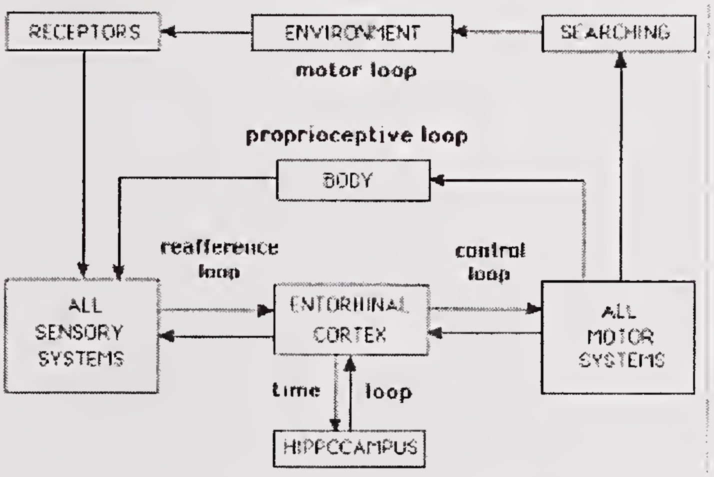


Figure 2. This diagram of brain state space maps the multiple feedback loops that support the intentional arc. Flow of neural activity inside the brain is in two directions. Forward flow from the sensory systems to the entorhinal cortex and on to the motor systems is by spatial AM patterns of action potentials at the **microscopic** level, by which transmitting cortices drive the neurons in their targets. Feedback flow from the motor systems to the entorhinal cortex by control loops, and from the entorhinal cortex to the sensory systems inside the brain, is by spatial AM patterns of action potentials at the **mesoscopic** level. This feedback constrains and modulates the microscopic activity in the forwardly transmitting populations. The mesoscopic feedback messages are order parameters that bias the attractor landscapes of the sensory cortices in preafference. Forward flow supports motor output and provides the content of percepts. Feedback flow supports integrative processes in learning that lead to the wholeness of intentionality. They enable the formation of a **macroscopic** AM pattern that reflects the integration of the activity of an entire hemisphere.

Owing to the nonlinear state transitions by which they form, these mesoscopic brain states are not representations of stimuli, nor are they simple effects caused by stimuli. Each learned stimulus serves to elicit the construction of a pattern that is shaped by the synaptic modifications between cortical neurons from prior learning, which vastly outnumber the synapses formed by incoming sensory axons, and also by the brain stem nuclei that bathe the forebrain in neuromodulatory chemicals. Each cortical activity pattern is a dynamic operator that creates and carries the meanings of stimuli for the recipient animal. It reflects the individual history, present context, and expectancy, corresponding to the unity and the wholeness of the intentionality. The patterns created in each cortex are unique to each animal. All sensory cortices transmit their signals into the limbic system, where they are integrated with each other over time, and the resultant integrated meaning is transmitted back to the cortices in the processes of selective attending, expectancy, and the prediction of future inputs, which together comprise the neural process of "reafference".

The same kinds of EEG activity as those found in the sensory and motor cortices are found in various parts of the limbic system. This discovery indicates that the limbic system also has the capacity to create its own spatiotemporal patterns of neural activity. They are related to past experience and convergent multisensory input, but they are self-organized. The limbic system provides a neural matrix of interconnections, that serves to generate continually the neural activity that forms goals and directs behavior toward them. EEG evidence shows that the process occurs in discontinuous steps, like frames in a motion picture. Each step follows a dynamic state transition, in which a complex assembly of neuron populations jumps suddenly from one spatiotemporal pattern to the next, as the behavior evolves. Being intrinsically unstable, the limbic system continually transits across states that emerge, spread into other parts of the brain, and then dissolve to give rise to new ones, a process that Japanese mathematicians have described as "chaotic itinerancy" between "attractor ruins" [34]. Its output controls the brain stem nuclei that serve to regulate its own excitability levels, implying that it regulates its own neurohumoral context, enabling it to respond with equal facility to changes that call for arousal and adaptation or rest and recreation, both in the body and the environment. It may be said that the neurodynamics of the limbic system, assisted by other parts of the forebrain such as the frontal lobes, initiates the novel and creative behavior seen in search by trial and error.

The limbic activity patterns of directed arousal and search are sent into the motor systems of the brain stem and spinal cord. Simultaneously, patterns are transmitted to the primary sensory cortices, preparing them for the consequences of motor actions. This process has been called "reafference" [12, 35], "corollary discharge" [32], "focused arousal" [29] and "preafference" [22, 23]. It sensitizes sensory systems to anticipated stimuli prior to their expected times of arrival Sensory cortical constructs consist of brief staccato messages to the limbic system, which convey what is sought and the result of the search. After multisensory convergence, the spatiotemporal activity pattern in the limbic system is up-dated through temporal integration in the hippocampus. Between sensory messages there are return up-dates from the limbic system to the sensory cortices, whereby each cortex receives input that has been integrated with the output of the others, reflecting the unity of intentionality. Everything that a human or an animal knows comes from this iterative circular process of action, reafference, perception, and up-date. It is done by successive frames that involve repeated state transitions and self-organized constructs in the sensory and limbic cortices. This neurodynamic system is defined here as the "limbic self" in the brain of an individual, where intentional behavior is created, with help from other parts of the forebrain.

An act of observation comprises Aquinas' intentional action of "stretching forth" and learning from the consequences. It embodies the existential "action-perception cycle" of Merleau-Ponty [26]. It corresponds to Piaget's [27]cycle of "action, assimilation, and adaptation" in the sensorimotor stage of childhood development. His postulated sequences of equilibrium, disequilibrium, and re-equilibration conform to state transitions in brain dynamics, which initiate and sustain action, construct dynamic patterns in the sensory cortices, and up-date the limbic patterns by modifying synapses in the learning that follows the sensory consequences of intended actions. For Piaget, cause and effect are chains of events that have the appearance of linkage corresponding to the unfolding experience of that exploration, by which a child is trying to make sense of its world by manipulating objects in it. The origin of causal inference is buried deeply in the pre-linguistic exploratory experience of each of us. It is not easily accessed by cognitive analysis or introspection.

We are all aware of our acts of observation. It is partly by expectation of what we are looking for through reafference, partly by perceiving the changes that our actions make in the dispositions of our bodies through proprioception, and partly by our selection of stimuli from the environment through exteroception. We perceive our intentional acts as the "causes" of changes in our perceptions, and the subsequent changes in our bodies as "effects" [12]. If this hypothesis of limbic dynamics is correct, then everything that we know we have learned through the action-perception cycle, including the iterative state changes by which it is produced in brains of animals and humans. It is this cycle, in prototypic form without need for appeal to consciousness, that must be simulated in our attempts to devise intelligent machines.

2.0 Characteristics of brain states as they are revealed by EEGs

The "state" of the brain is a description of what it is doing in some specified time period. A state transition occurs when the brain changes and does something else. For example, locomotion is a state, within which walking is a rhythmic pattern of activity that involves large parts of the brain, spinal cord, muscles and bones. The entire neuromuscular system changes almost instantly with the transition to a pattern of jogging or running. Similarly, a sleeping state can be taken as a whole, or divided into a sequence of slow wave and REM stages. Transit to a waking state can occur in a fraction of a second, whereby the entire brain and body shift gears, so to speak. The state of a neuron can be described as active and firing or as silent, with sudden changes in the firing manifesting state transitions. Populations of neurons also have a range of states, such as slow wave, fast activity, seizure, or silence. The mathematics of nonlinear dynamics is designed to study these states and the transitions by which they are accessed and abandoned.

2.1 The problem of stability of cortical states

The most critical question to ask about a state is its degree of stability or resistance to change. Evaluation is done by perturbing an object or a system [8]. For example, an object like an egg on a flat surface is unstable, but a coffee mug is stable. A person standing on a moving bus and holding on to a railing is stable, but someone walking in the aisle is not. If a person regains his chosen posture after each perturbation, no matter in which direction the displacement occurred, that state is regarded as stable, and it is said to be governed by an attractor. This is a metaphor to say that the system goes ("is attracted") to the state through an interim state of transience. The range of displacement from which recovery can occur defines the basin of attraction, in analogy to a ball rolling to the bottom of a bowl. If the perturbation is so strong that it causes concussion or a broken leg, and the person cannot stand up again, then the system has been placed outside the basin of attraction, and a new state supervenes with its own attractor and basin.

Stability is always relative to the time duration of observation and the criteria for what is chosen to be observed. In the perspective of a lifetime, brains appear to be highly stable, in their numbers of neurons, their architectures and major patterns of connection, and in the patterns of behavior they produce, including the character and identity of the individual that can be recognized and followed for many years. Brains undergo repeated transitions from waking to sleeping and back again, coming up refreshed with a good night or irritable with insomnia, but still, giving the same persons as the night before. Personal identity is usually quite stable. But in the perspective of the short term, brains are highly unstable. Thoughts go fleeting through awareness, and the face and body twitch with the passing of emotions. Glimpses of their internal states of neural activity reveal patterns that are more like hurricanes than the orderly march of symbols in a computer. Brain states and the states of populations of neurons that interact to give brain function, are highly irregular in spatial form and time course. They emerge, persist for a small fraction of a second, then disappear and are replaced by other states. It is the flexibility and creativeness of this process that makes it so successful in animals for their adaptation to rapidly changing and unpredictable environments, and that makes it the desired platform on which to base the design of intelligent machines.

2.2 Three types of stable cortical states

In using dynamics we approach the problem by defining three kinds of stable state, each with its type of attractor. The simplest is the point attractor. The system is at rest unless perturbed, and it returns to rest when allowed to do so. As it relaxes to rest, it has the history of what happened, but that history is lost after convergence to rest. Examples of point attractors are silent neurons or neural populations that have been isolated from the brain, and also the brain that is depressed into inactivity by injury or a strong anesthetic, to the point where the EEG has gone flat (Figure 3, bottom trace). A special case of a point attractor is noise. This state is observed in populations of neurons in the brain of a subject at rest, with no evidence of overt behavior. The neurons fire continually but not in concert with each other. Their pulses occur in long trains at irregular times. Knowledge about the prior pulse trains from each neuron and those of its neighbors up to the present fails to support the prediction of when the next pulse will occur. The state of noise has continual activity with no history of how it started, and it gives only the expectation that its amplitude and other statistical properties will persist unchanged.

A system that gives periodic behavior is said to have a limit cycle attractor. The classic example is the clock. When it is viewed in terms of its ceaseless motion, it is regarded as unstable until it winds down, runs out of power, and goes to a point attractor. If it resumes its regular beat after it is re-set or otherwise perturbed, it is stable as long as its power lasts. Its history is limited to one cycle, after which there is no retention of its transient approach in its basin to its attractor. Neurons in populations rarely fire periodically, and when they appear to do so, close inspection shows that the activities are in fact irregular and unpredictable in detail, and when periodic activity does occur, it is either intentional, as in rhythmic drumming, clapping and dancing, or it is pathological, as in the periodic oscillations of the eyes in nystagmus, or of the limbs during Parkinsonian tremor, or of the cortex during the hypersynchrony of partial complex seizures that are revealed by near-periodic spike trains (Figure 3, top trace).

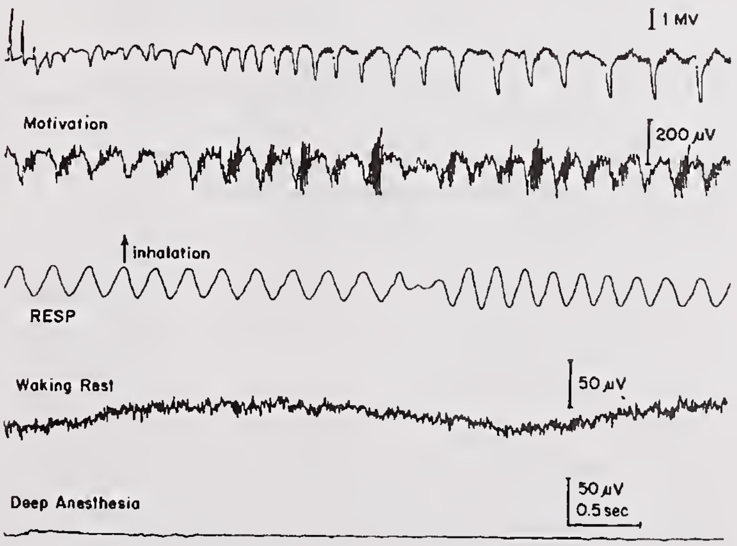


Figure 3. Four levels of function of the olfactory system are revealed by EEG recording. The lowest is the non-interactive 'open loop' state imposed by deep anesthesia, which suppresses brain activity. The next is the resting steady state with broad spectrum $1/f^2$ aperiodic waves. The aroused level in which behavior is generated is shown by the repeated state transitions, by which bursts are formed that reveal spatial patterns of AM (amplitude modulation) relating to odorant recognition with inhalation. The upper trace shows the pattern of high-amplitude spikes when an epileptic seizure has been triggered by powerful electrical stimulation. This state is likewise chaotic, but with a reduced correlation dimension. This state also occurs during recovery from deep anesthesia on the way to the resting state [9, 31].

The third type of attractor gives aperiodic oscillation of the kind that is observed in recordings of EEGs. There is no one or small number of frequencies at which the system oscillates. The system behavior is therefore unpredictable, because performance can only be projected far into the future for periodic behavior. This type is now widely known as "chaotic". The existence of this type of oscillation was known to Poincaré a century ago, but systematic study was possible only recently after the full development of digital computers. The best known systems with chaotic attractors have a small number of components and a few degrees of freedom, as for example, the double-hinged pendulum, the dripping faucet, and the Lorenz, Chua, and Rössler attractors [13]. These simple models are stationary, autonomous, and noise-free, forming the class of "deterministic chaos". Large and complex real-world systems, which include neurons and neural populations are noisy, infinite-dimensional, nonstationary, non-autonomous, yet capable of chaotic behavior which has been called "stochastic chaos" [14]. The source is postulated to be the synaptic interaction of millions of neurons, which create fields of microscopic noise in cortex, but which are constrained by their own interactions to generate mesoscopic order parameters that regulate the spatiotemporal patterns of cortical activity revealed by the EEG. These spatiotemporal patterns are revealed by spatial patterns of amplitude modulation ("AM patterns") of a spatially coherent aperiodic carrier wave in the gamma range of the EEG. They appear in time series as bursts of oscillation (Figure 3), and their spatial patterning indicates the existence of an attractor landscape, which is actualized in the olfactory system with each inhalation (Figures 4 and 5 during intentional behavior.

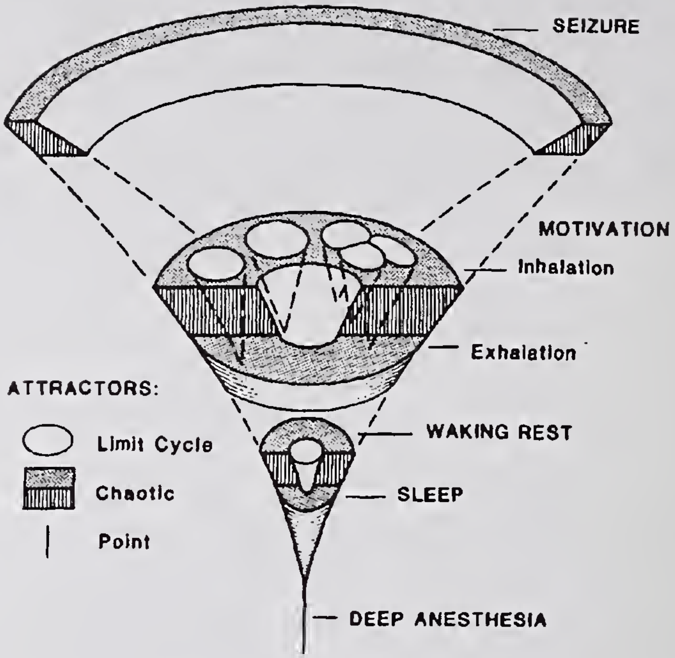


Figure 4. A bifurcation diagram of the olfactory system state space is constructed from the EEGs in Figure 3.

The discovery that brain dynamics operates in chaotic domains has profound implications for the study of higher brain function [31]. A chaotic system has the capacity to create novel and unexpected patterns of activity. It can jump instantly from one mode of behavior to another, which manifests the facts that it has a collection of attractors, each with its basin, and that it can move from one to another in an itinerant trajectory [34]. It retains in its pathway across its basins its history, which fades into its past, just as its predictability into its future decreases. Transitions between chaotic states constitute the dynamics that we need to understand how brains perform such remarkable feats as abstraction of the essentials of figures from complex, unknown and unpredictable backgrounds, generalization over examples of recurring objects never twice appearing the same, reliable assignment to classes that lead to appropriate actions, and constant up-dating by learning.

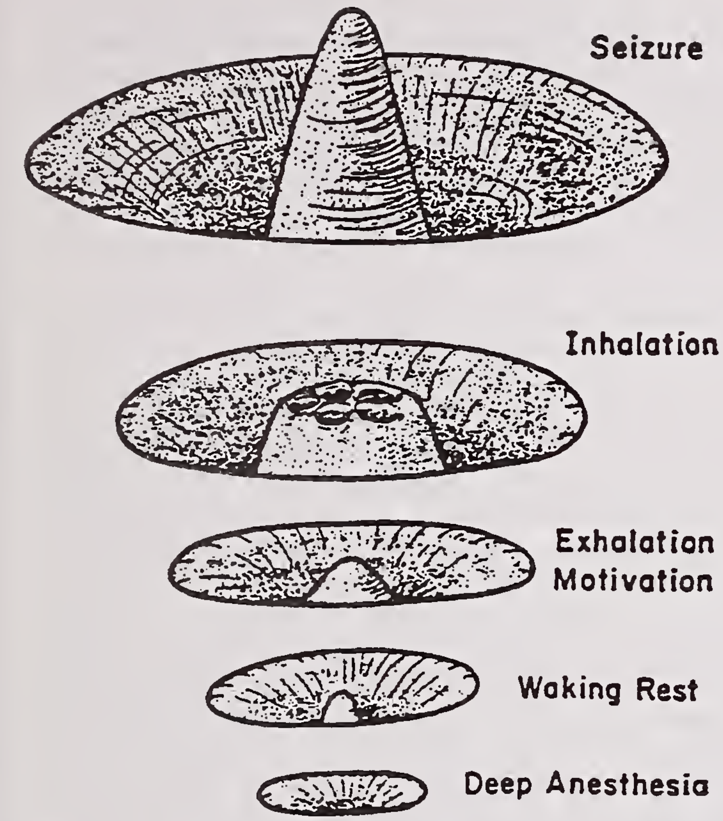


Figure 5. This perspective drawing of a projection from an infinite dimensional brain state space into 3-space offers a view of how an attractor landscape of learned basins of attraction is created with each inhalation. The selection is made by the input odorant. If the stimulus is novel or unknown, the system goes into the chaotic well, which provides the aperiodic unpatterned activity that drives Hebbian learning for new basin formation.

2.3 The 1st order cortical state transition is an elemental step in intention

Systems such as neurons and brains that have multiple chaotic attractors also have point and limit attractors, each with its basin of attraction, which serves to provide the generalization gradient required for perception of recurring stimuli that are never twice the same. If the basin is that of a point or a limit cycle attractor, the system can proceed predictably to an identical end state. If the basin leads to a chaotic attractor, the system goes into ceaseless fluctuation, as long as its energy lasts. If the starting point is identical on repeated trials, which can only be assured by simulation of the dynamics on a digital computer, the same aperiodic behavior appears. If the starting point is changed by an arbitrarily small amount, although the system is still in the same basin, the trajectory is not identical. A deterministic chaotic system that is in the basin of one of its chaotic attractors is legendary for its sensitivity to the initial conditions. If the difference in starting conditions is too small to be originally detected, it can be inferred from the unfolding behavior of the system, as the difference in trajectories becomes apparent. This observation shows that a chaotic system has the capacity to create novel patterns constituting endogenous increases in information in the course of continually constructing its own trajectory into the future.

Our EEG evidence indicates that every primary sensory cortex maintains multiple basins corresponding to previously learned classes of stimuli, as well as to the unstimulated state, which together form an attractor landscape. They all show evidence that the vehicle they use for transmission of their output is an aperiodic carrier wave that is amplitude-modulated in the two spatial dimensions of cortical coding, and that is gated by extra-cortical forcing functions in the theta range (2-7 Hz). We note that we predicted a common code for all sensory systems, on the basis that the signals from all sensory cortices must be combined in the limbic system to form gestalts. We postulate that preafferent input from the limbic system can serve to bias the landscapes in such a way as to facilitate the capture of the multiple sensory systems by basins of the attractors corresponding to the goal of the intended observation, perhaps in the manner of the variable tiling in a Voronoi diagram. This chaotic prestimulus state of expectancy establishes the sensitivities of the cortices, so that the very small number of sensory action potentials evoked by the expected stimuli can simultaneously carry the cortical trajectories into the basins of the appropriate attractors as they are created by the forcing function, in the case of olfaction by inhalation (Figure 5), irrespective of which equivalent receptors actually receive the expected stimuli in the different sensory modalities. In the absence of the stimulus, the cortices continue to transmit their outputs to the limbic system, confirming the continuing absence. The stimuli are also selected by the limbic system through orientation of the sensory receptors in space by sniffing, looking, and listening. We believe that the basins of attraction in each of the sensory cortices are shaped by limbic input to sensitize them for receiving and processing the desired class of stimuli in every modality, whatever may be the goal at the moment of choice.

3.0 Problems in use of chaotic dynamics in the development of advanced machine intelligence

Chaotic dynamics has proved to be extremely difficult to harness in the service of intelligent machines. Most studies that purport to control chaos either find ways to suppress it and replace it with periodic or quasiperiodic fluctuations, or to lock two or more oscillators into synchrony sharing a common aperiodic wave form, often as an optimal means for encryptation and secure transmission. Our aim is to employ chaotic dynamics as the means for creating novel and endogenous space-time patterns, which must be the means to achieve any significant degree of autonomy in devices that must operate far from human guidance, where in order to function they must make up their courses of action as they go along. We know of no other way to approach a solution to the problem of how to introduce creative processes into machines, other than to simulate the dynamics we have found in animal brains. To be sure, there are major unsolved problems in this approach, chief among them that we know too little about the dynamics of the limbic system. Hence we find it necessary to restrict the development of hardware models to the stage of brain-world interaction that we know best, which is the field of perception. In brief, what are the problems in giving eyes, ears and a nose to a robot, so that it might learn about its environment in something like the way that even the simpler animals do - by creating hypotheses and testing them through their own actions?

3.1 Noise stabilization of chaotic dynamics, opening the way to analog-digital hybrid embodiments

The operations in the olfactory system by which the state transitions and pattern constructions for pattern classification are simulated in software and hardware embodiments have been described in a series of publications [9-12, 14]. Our simulations are done with a set of approximately 920 interconnected first-order nonlinear ordinary differential equations, forming what we have named the KIII model [8]. The basic element, the KO set, is a 2-stage linear integrator simulated in hardware [6, 7] by 2 operational amplifiers, whose output is passed through an asymmetric sigmoid function modeled by 2 diodes back-to-back. Connections between 64 elements are time multiplexed (Figure 6) through a MUX, an amplifier with voltage-controlled gain, and a DMUX [10]. Switching is controlled by a digital computer at a clock rate suitable for the pass band of the carrier wave. For each connected pair the gain is stored in memory, so that the connection strengths are easily modified during learning. With this device the connectivity grows by 2-N instead of by N^{2}`. In digital embodiment the equations have been solved by numerical integration on Unix, Macintosh, and PC platforms, and by vector programming on the Cray M/X.

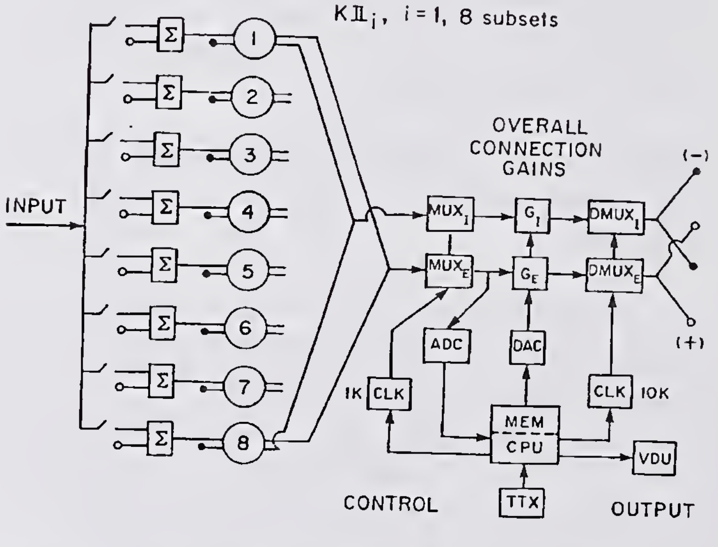


Figure 6. Schematic for connecting KII sets by multiplexing.

Interaction of KO sets of like kind (excitatory or inhibitory) giving point attractors is modeled by KI sets; interaction of KIe and KIi sets giving limit cycle attractors is modeled by KII sets. Three serial KII sets in layers that correspond to the olfactory bulb, prepyriform cortex, and an intervening control nucleus called the AON is modeled by the KIII set; if the 3 characteristic frequencies are incommensurate, and the feedback delays between the 3 layers are distributed to act as low-pass filters, the solutions of the equations give the aperiodic waveforms and broad $1/f^{2}$ spectra (Figure 7) of EEGs trom the 3 layers. The asymmetric sigmoid endows the system with the property of nonlinear state transitions on step inputs, owing to the amplitude-dependent gain of the KO elements.

In the course of digital simulation it has become apparent that a minimum of 64 elements will suffice for 2-D pattern classification under Hebbian and non-Hebbian reinforcement learning [16, 24, 25, 37, 38]. The large number of equations leads to attractor crowding [15], in which the basins of attraction shrink close to the size of the digitizing step in using rational numbers for computation, so that sooner or later the system jumps out of its designated chaotic basin into a neighboring basin that is most likely to be that of a point or limit cycle attractor, which kills the system. This problem has been solved by use of additive noise on the order of 15% of the amplitude of the aperiodic state variables [4, 5, 13, 15], giving robust attractor landscapes for learning and pattern classification [24, 25]. The lesson learned is that deterministic chaos, in which the system is low-dimensional, stationary, strictly autonomous, and noise-free, is inappropriate for modeling biological and machine intelligence. Brains operate with what we call 'stochastic chaos' [13], which is high-dimensional, nonstationary with regularly repeated state transitions, engaged with its surround, and deeply embedded in noise created by KIe sets and manifested in high densities of action potentials. The noise in digital models is simulated with random number generators, either rectified to simulate KIe sets or off-set with d.c. bias to simulate the noise in KIIei sets.

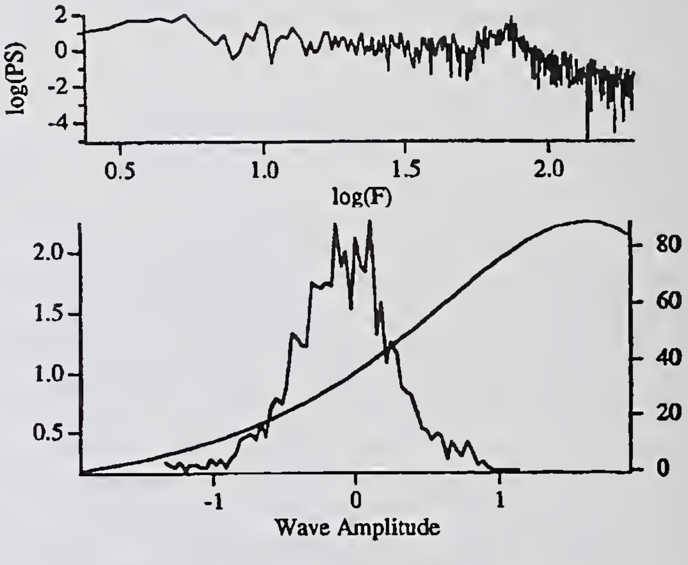


Figure 7. The power spectrum and amplitude histogram for a simulated EEG trace from the KIII model, with a section of the asymmetric nonlinear gain curve, showing the nature of the nonlinearity that provides for destabilization by the input. The interactive gain increases with excitatory input.

The finding in digital embodiments that noise is not only unavoidable but is necessary for stable high-dimensional chaotic dynamics opens the way to analog embodiments [7], in which noisy components resemble the characteristics of local pools in nerve cell assemblies, but which offer much higher rate of temporal and spatial integration, the use of continuous variables in place of rational numbers, and the feasibility of implementing the dynamics on chips suitable for incorporation into mobile devices.

3.2 Embedding devices for perception into autonomous cognitive machines

The KII sets have multiple robust limit cycle attractors, which become embedded as chaotic attractors when coupled in serial layers with distributed delayed feedback. The KIII model is offered as the prototype for constructing devices in hardware and software to implement the elementary steps of perception, thus providing robots with the sensory ports that they need to guide them through their environments. These steps are the interpretive operations necessary to normalize, compress, abstract and generalize over successive inputs preparatory to classification [5, 16, 25, 33, 36]. These cognitive operations are done by the nonlinear operations in the input stage and by the basins of attraction in the landscape formed by learning in each of the sensory systems. They are required in each of the ports providing information to the mobile device about its visual, auditory, tactile, and chemical environments. Our tests of the KIII model have shown that it can learn a new class in half a dozen trials instead of the thousands of trials required by MLPs, and that new learning occurs without degradation of previous attractors, although, as in the case of the olfactory system, the attractors are modified through attractor crowding. The superior level of 'intelligence' is demonstrated by the capacity of the KIII model to separate items in 64-space that belong to identifiable classes but are not linearly separable. The classes are, in fact, constructed by the model and are not imposed from outside, constituting an aspect of autonomy. In other words, the system creates its own features from its own experience of the constancy of relations between channels in the 8x8 64-channel input array.

Formation of a world-view by which the device can guide its explorations for the means to reach its goals depends on the integration of the outputs of the several sensory systems, in order to form a multisensory percept known as a gestalt. This integration is easily done when all of the ports have their outputs in the samc form: a vector consisting of a 2-D spatial pattern of amplitude modulation of a 1-D aperiodic wave form in the gamma range (nominally 30-60 Hz), which is segmented in time at a frame rate of nominally 2-7 Hz and frame durations on the order of 0.1 sec. Precise clocking and synchronization are not prerequisite.

The sequential frames deriving from sampling the environment must then be integrated over time and oriented in space. An example of how these higher operations might be done was provided by W. Gray Walter [36] with his electronic tortoises, which had the capacity for autonomous goal-directed search involving the adjudication of conflicting needs in an uncertain environment.

The performances of these devices set a challenging level of 'intelligence' to which to aspire, and they also serve to highlight some of the difficulties in using the descriptive term "autonomous". As with animals the devices were untethered, and they learned to avoid obstacles without need for instruction or intervention, if within their limited capacities for locomotion. However, they were programmed to satisfy their own needs without regard for or comprehension of anything else's, perhaps in analogy to house pets, whose sole purpose, however inadvertent, is to provide enjoyment to their owners, and seldom to do useful work or bend their talents to the benefits of the owners, or, in the case of the machines, the designers and builders.

It is already apparent that fully autonomous vehicles are not in the best interest of researchers and the general public, except as demonstrations of what might emerge as major problems from this line of study. It is also clear that such devices can and will be built, and that the proper path of future management will not be by techniques of training and aversive conditioning, but by education, with inculcation of desired values determined by the manufacturers that will govern the choices that must by definition be made by the newly autonomous mechanical devices.

Acknowledgments

This work was supported by grants from the National Institute of Mental Health MH06686, the Office of Naval Research N00014-90-J-4054, and from ARO MURI DAAH04-96-1-0341.

References

1] Aquinas, St. Thomas. (1272). *Treatise on Man.* In: Summa Theologica. Translated by Fathers of the English Dominican Province. Revised by Daniel J Sullivan. Published by William Benton as Volume 19 in the Great Books Series. Chicago: Encyclopedia Britannica, Inc., 1952.

2} Barrie JM, Freeman WJ, Lenhart M (1996) Modulation by discriminative training of spatial patterns of gamma EEG amplitude and phase in neocortex of rabbits. Journal of Neurophysiology 76: 520-539.

3] Bures J, Buresová O, Krivánek J (1974) The Mechanism and Applications of Leão's Spreading Depression of Electroencephalographic Activity. New York: Academic Press.

4] Chang H-J, Freeman WJ (1998) Biologically modeled noise stabilizing neurodynamics for pattern recognition. International Journal of Bifurcation & Chaos 8 (2), 321-345.

5] Chang H-J, Freeman WJ (1999) Local homeostasis stabilizes a model of the olfactory system globally in respect to perturbations by input during pattern classification. International Journal of Bifurcation and Chaos 8: 2107-2123.

6] Edelman JA, Freeman WJ (1990) Simulation and analysis of a model of mitral-granule cell population interactions in the mammalian olfactory bulb. Proceedings IJCNN I: 62-65.

7] Eisenberg J, Freeman WJ, Burke B (1989) Hardware architecture of a neural network model simulating pattern recognition by the olfactory bulb. Neural Networks 2: 315-325.

8] Freeman WJ (1975) Mass Action in the Nervous System. New York: Academic Press.

9] Freeman WJ (1987) Simulation of chaotic EEG patterns with a dynamic model of the olfactory system. Biological Cybernetics 56: 139-150.

10] Freeman WJ (1988) Pattern learning and recognition device. United States Patent # 4, 748, 674, May 31, 1988.

11] Freeman WJ (1992) Neurons to Brain Chaos. International Journal of Bifurcation and Chaos 2: 451-482.

12] Freeman WJ (1995) Societies of Brains. Mahwah NJ, Lawrence Erlbaum Associates.

13] Freeman WJ (1999) Noise-induced first-order phase transitions in chaotic brain activity. International Journal of Bifurcation and Chaos 9: 2215-2218.

14] Freeman WJ [2000] Neurodynamics. An Exploration of Mesoscopic Brain Dynamics. London UK: Springer-Verlag.

15] Freeman WJ, Chang H-J, Burke BC, Rose PA, Badler J (1997) Taming chaos: Stabilization of aperiodic attractors by noise. IEEE Transactions on Circuits and Systems 44: 989-996.

16] Freeman WJ, Yao Y, Burke B. (1988) Central pattern generating and recognizing in olfactory bulb: A correlation learning rule. Neural Networks 1: 277-288.

17] Goltz FL (1892) Der Hund ohne Grosshirn. Siebente Abhandlung über die Verrichtungen des Grosshirns. Pflügers Archiv 51: 570-614.

18] Gray CM (1994) Synchronous oscillations in neuronal systems: mechanisms and functions. Journal of Comparative Neuroscience 1: 11-38.

19] Haken H (!983) Synergetics: An Introduction. Berlin: Springer.-Verlag

20] Hardcastle VG (1994) Psychology's binding problem and possible neurobiological solutions. Journal of Consciousness Studies 1: 66-90.

21] Herrick CJ (1948) The Brain of the Tiger Salamander. Chicago IL: University of Chicago Press.

22] Kay LM, Freeman WJ (1998) Bidirectional processing in the olfactory-limbic axis during olfactory behavior. Behavioral Neuroscience 112: 541-553.

23] Kay LM, Lancaster L, Freeman WJ (1996) Reafference and attractors in the olfactory system during odor recognition. International Journal of Neural Systems 7: 489-496.

24] Kozma R, Freeman WJ (1999) A possible mechanism for intermittent oscillations in the KIII model of dynamic memories - the case study of olfaction. Proceedings, IJCNN'1999, Washington DC.

25] Kozma R, Freeman WJ (2000) Encoding and recall of noisy data as chaotic spatio-temporal memory patterns in the style of the brains. Proceedings, ICJNN'2000. Como, Italy

26] Merleau-Ponty M (1942) The Structure of Behavior (AL Fischer, Trans.). Boston: Beacon Press (1963).

27] Piaget J (1930) The child's conception of physical causality. New York: Harcourt, Brace. p. 269

28] Roth G (1987) Visual Behavior in Salamanders. Berlin: Springer-Verlag

29] Sheer DE (1989) Sensory and cognitive 40-Hz event-related potentials: Behavioral correlates, brain function, and clinical application. Brain Dynamics. Basar E, Bullock TH (eds.) Berlin: Springer-Verlag.

30] Singer W, Gray CM (1995) Visual feature integration and the temporal correlation hypothesis. Annual Review of Neuroscience 18: 555-586.

31] Skarda CA, Freeman WJ (1987) How brains make chaos in order to make sense of the world. Behavioral and Brain Sciences 10: 161-195.

32] Sperry RW (1950) Neural basis of the spontaneous optokinetic response. Journal of Comparative Physiology 43: 482-489.

33] Storm C, Freeman WJ (1999) A novel dynamical invariant measure addresses the stability of the chaotic KIII neural network. Proceedings, IJCNN'1999, Washington DC.

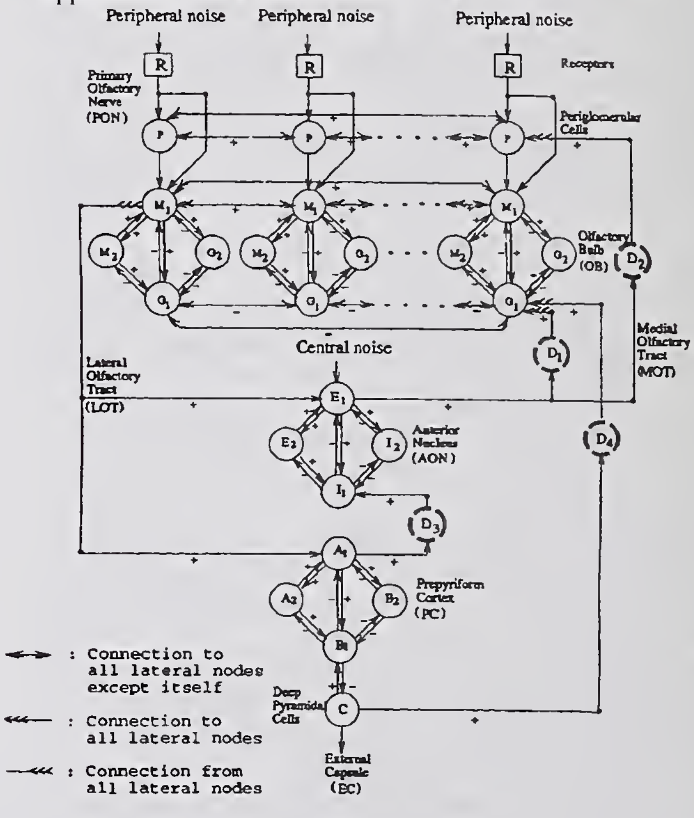
34] Tsuda I (1996) A new type of self-organization associated with chaotic dynamics in neural networks. International Journal of Neural Systems 7: 451-459.

35] von Holst E & Mittelstaedt H (1950) Das Reafferenzprinzip Naturwissenschaften 37: 464-476.

36] Walter WG (1963) The Living Brain. New York: Norton.

37] Yao Y, Freeman WJ (1990) Model of biological pattern recognition with spatially chaotic dynamics. Neural Networks 3: 153-170.

38] Yao, Y., Freeman WJ, Burke, B., Yang, Q. (1991) Pattern recognition by a distributed neural network: An industrial application. Neural Networks 4: 103-121.



Measuring intelligence: a neuromorphic perspective

Marwan A. Jabri
Electrical and Computer Engineering
Oregon Graduate Institute, 20000 NW Walker Rd,
Beaverton, OR 97006 USA
and
School of Electrical and Information Engineering
The University of Sydney, NSW 2006 Australia
marwan@ece.ogi.edu

ABSTRACT

Neuromorphic engineering is about the development of biologically inspired roving machines that can exhibit intelligent behaviour, learn on-line and in real-time. The question of how to assess and measure the intelligence of such machines is essential if progress in neuromorphic engineering is to be assessed. However, it is awkward to talk about measuring intelligence without a clear understanding of the capabilities that researchers aim or dream to equip neuromorphic systems with. In this communication we promote the position that metrics for measuring of the intelligence of neuromorphic systems should be task-based, should factor in the computational resources, the on-line learning efficiency, the capability to learn from intermittent reward that can vary in frequency and importance to the task at hand, the capability to anticipate events and to modify decision making processes based on anticipated events, the capability to balance exploration and exploitation as to discover new methods or to fine-tune existing methods, and the ability to optimize the utilization of its resources using ground rules that maximizes it success. To factor in all these aspects requires a fundamental assessment of what such machines achieve as goals and at what cost. We propose that a simple achievement rule, energy and resource oriented metric be used.

Keywords: *neuromorphic engineering, on-line learning, reward-based learning, anticipation, exploration and exploitation, regularity and modularity.*

1 Introduction

Neuromorphic engineering was a term coined by Carver Mead and described the process of building systems based on biological models and embedding them in roving machines. In the last 20 years, neuromorphic engineering addressed the development of various biological like processing system such as retinas, cochleas (Schaik 2000), legged robots and creatures (Tilden 1994) (Lewis, Etienne-Cummings et al. 2000), sensorimotor control (Horiuchi and Koch 1999) (Etienne-Cummings, Spiegel et al. 2000) and integration systems (Jabri, Coenen et al. 1997).

Although analog microelectronics was initially promoted (and continue to some extent) as the ideal substrate for neuromorphic information processing systems (Mead 1989), current works tend to use many implementation technologies, hardware and software.

The aim of many neuromorphic engineering groups is to develop active perception systems, systems that interact with the environment in a closed loop fashion[1].

Neuromorphic engineering is a synergy between neuroscience and engineering. The common neuromorphic methodology is to identify a task or a function, to explore and identify brain areas from neuroscientific knowledge (anatomy, physiology,

[1] The Telluride Neuromorphic Engineering workshop is a yearly meeting where research groups meet and collaborate. See http://zig.ini.unizh.ch/telluride2000.

psychophysics, ...), to develop computational models that encapsulate the information processing at some level of abstraction, and to develop implementations of the models. The determination of an acceptable level of abstraction of the biological systems during the computational model development is a challenging task, and is typically done as to preserve some essence of the biological information or mechanical processes.

In assessing the intelligence of engineering machines, and because of its close relationship to the neuroscientific community, neuromorphic engineering has traditionally relied on several levels of metrics. Not all metrics are necessarily directly related to the behaviour of the machines and the classification of the intelligence of such behaviour. The common levels are:

- Device/circuits
- Representation
- Organization
- Behaviour with and without artificial lesions
- Learning & behaviour adaptation

In these assessments, tasks have commonly been related to the biological systems being modeled: specific brain areas, the central nervous system, and the mechanical apparatus. We elaborate on these tasks in the next section.

2 Neuromorphic Tasks – Present and Future

2.1 Peripherals systems

Biologically based or inspired peripheral systems are probably the most researched neuromorphic systems. The development of silicon-based implementations of retinas and cochleas has been pioneered in Carver Mead's laboratory in the eighties. Artificial olfactory and somatosensory systems have also been researched and developed.

It is clear why most early neuromorphic research focused on peripheral systems: They are the sensors and they drive the motor responses of biological systems, and they are the most understood, in particular in the case of primates.

The research and development of neuromorphic peripheral systems has also contributed to better understanding of the biological devices and they incorporation in systems.

2.2 Sensorimotor Systems

Over the last decade sensorimotor systems have been developed. Sensorimotor system are broad in their definition but are supposed to implement forms of sensory (visual, auditory, infrared, sonar) to motor mapping, where the motor are actions that aim at performing forms of active perception, navigation and tracking, or orienting to stimuli in the environment.

Experimental sensorimotor systems have incorporated abilities by incorporating simplified models of the superior colliculus, goal reaching and simple navigation abilities by incorporating computational models of the basal ganglia and ventral tegmental area, predictive control abilities by incorporating computation models of the cerebellum, and spatial representation learning by incorporating computational models of the hippocampal formation.

Sensorimotor systems have so far included sophisticated adaptive learning abilities implemented in software. The measuring of the intelligence of such systems has largely been a matter of retrieving behavioral properties that resemble those of animal, when the systems are implementing sensorimotor tasks, or by observing the behaviour of the system when software lesions are performed. In that case the deficit of the systems are typically compared to those of animal that have had specific brain areas severed or temporarily disabled.

Analog Very Large Scale Integration (aVLSI) systems with adaptive abilities have also been implemented and typically mimic to some extent their biological counter parts. The assessment of the intelligence of such system has largely been a matter of comparing the signal processing or collective computation of the devices to the biological counterparts. These could also be seen as task-oriented comparison. An example is an implementation of the retina with adaptive intensity saturation control. Another example is a silicon cochlea that implements adaptive gain control.

2.3 Cognitive Systems

If one defines cognitive systems as being capable of performing higher order processing by utilizing first order information and generating higher order knowledge, some sensorimotor systems would qualify of being “cognitive”.

An example is a sensorimotor system similar to that of Fig 1 which implements abstract computational of the cerebellum, basal ganglia and ventral tegmental area.

In this system the cerebellum performs sensory prediction and coordinate transformation from world coordinate to robot centered coordinate. The predicted sensory signal (visual target position) is then used by the basal ganglia to associate a motor command with the visual target as to keep it as much as possible in front of it. If the basal ganglia are lesioned, the robot looses its motor ability. If the cerebellum is lesioned, the tracking lags the object.

In survival terms, and such a sensorimotor system is controlling the hunting abilities of an animal, a lesion of the cerebellum would most likely lead to the animal death, although it can track its prey, though not predictively to the point that it can catch it (assuming a mobile prey), or it cannot escape a predator by anticipating potential contact points. Interestingly, the hypothetical animal would be able to anticipate the position and perform all desired coordinate transformation, however a lesion of the basal ganglia will also lead to its death.

2.4 Future of Neuromorphic Systems

With the rapid development in neuroscience research brought by phenomenal growth in computation, sensing, signal processing and imaging technologies, neuromorphic engineering will increasingly focus on the implementation of complex motor, sensory and cognitive processing. The development of computational models of sub-cortical and cortical will permit the development of sophisticated real-time systems, that will go beyond present sensorimotor loops and will integrate aspects such as planning, object recognition, motion and auditory analysis, and perception. This will put additional pressure in comparing the performance of such systems, and hence on the issue of measurement metrics.

3 Computational Resources

Computational resources in neuromorphic systems, in particular aVLSI systems tend to be a central criterion of design. One attraction of aVLSI neuromorphic systems is the low power requirements (Mead 1989; Jabri, Coggins et al. 1996). However, beside the elegance of the implementation, and specific application requirements, it is becoming more difficult to promote analog as a preferred design methodology, except in some fairly narrow areas such as world interfaces. This is not to say that analog asynchronous parallel computation does not provide any conceptual computational advantages. Only that the inspirations for such advantages have not been met with clear theoretical support over digital computation as yet.

Computational resources have also been considered from the point of view of compactness, efficiency of representing basic computational elements such as sensors and signal processors. Here applications that have specific requirements such as ultra low power and high fault tolerance capability could benefit more from analog than digital representation. This is particularly the case if sparse representation is being used. In a sparse representation of neural networks, neurons within a hierarchy of computation do not fire concurrently. The receptive fields of the neurons are highly tuned/selective and are independent of each other. This translates into data-driven architecture with attractive low power consumption properties.

Given the infancy of neuromorphic systems, autonomous behaviour has not been developed beyond adaptive sensing and signal processing tasks.

4 On-line Learning

Continuous on-line learning with bound resources represents a challenge because of the following problems:

1- Frequency of the associations to be learnt is not sufficiently high to be captured in a distributed representation. Note the tuning pf learning parameters do not necessarily solve this problem as for example, the use of large learning rate can lead to prior information to be forgotten (catastrophic learning effects) if no processes are implemented to move and consolidate information from soft-term memory to long-term memory store.

2- In cases where statistical properties of the sensed signals are to be discovered on-line, sample size effects, and non-stationarity of the signals are very problematic. For example if independent component analysis techniques are being used to discover feature detectors (Bell and Sejnowski 1995), such discovery using information maximization techniques and mutual independence criteria of the features would be more difficult to achieve if performed on-line.

3- Rapid and flexible learning schedules is necessary in situations where autonomous systems requires to learn at various rates and in real-time. This imposes constraints on propagation of information in the system and on its time constants. For example systems doing sequence learning require significant memory resources in the form of analog or digital delay lines and the performance of credit assignment through time over the present and historical information.

The learning issues above represent significant challenges to the incorporation of online and continuous learning. The proposition of metrics for these sort of capabilities is premature, given we do not really know the how, when and where of such capabilities.

5 Anticipation

An important element in neuromorphic systems research is the development of the concept of anticipation within the context of autonomy. The system described earlier in Section 2.3 is an example demonstrating an anticipation property. A roving robot that can anticipate undesirable events would maximize its mission success. Anticipation or prediction of sensory or motor control (predictive control) have been attributed as a role to the cerebellum (Coenen and Sejnowski 1996; Coenen 1998), in addition to the traditional attributed role of motor learning (Marr 1969; Albus 1971).

Present computational models of the cerebellum have addressed individual sensory (or a few) and motor prediction capability. Computational models that demonstrate abilities to adaptively and continuously deal with a large number of sensory modality and motor learning skills are still to be developed. Such skills will be essential to autonomous machines that are expected to perform tasks such as navigation in complex terrains or to perform object manipulation. Anticipation is also important for planning because it affects the performance of the machine and its interaction with the environment and its objects. Measuring anticipation can be very subjective. However factoring anticipation in the overall goal of a machine will provide easier means for assessment.

6 Curiosity, Exploration and Exploitation

Autonomous machines should possess elements of "curiosity". For instance, it is known that reinforcement based learning algorithms depend on forms of exploration (Sutton and Barto 1981). However, exploration has so far been implemented in terms of probabilistic random actions aimed at exploring the state-space with hope of discovering policies that can be effective in achieving specific goals. The issues of either exploring more effectively or in a directed way, or to explore better policies and solutions are not well understood.

Another important aspect of autonomous systems is that of exploitation of infrequent, but yet important information encountered during machine experiences. The interactions between exploration and exploitation are fundamental in that regard. Reinforcement based learning algorithms have assumed that rewards are specified as end-achievements to the learning machine. The ability to discover and capture sensorimotor associations to yet unspecified goals (and reward) is essential to the rapid learning and the effective exploitation of sensorimotor experiences. To achieve this, the learning machines must be able to recognize unspecified or unscheduled rewards by forms of assessment of its sensory state and its sensory-reward memory.

7 Robustness and fault tolerance

Autonomous systems have to be robust and fault tolerant. We discussed in Section 3 sparse representation and their low power property as well as their potential role in more effective learning by decorrelating features. It is not clear however, without clear redundancy in the underlying resources (e.g. synapses and neurons), that sparse representation alone lead to more fault tolerance. It is also conceivable that other additional encoding representations, such as population-based be a source of fault tolerance (see for instance motor population coding (Georgopoulos 1995)).

Fault tolerance has been attributed to traditional neural network representation because of the distributed representation that develop during learning or that have been hand-crafted. The relationship between pure distributed representation and neural correlate is not trivial, nor automatic. Biological systems have various level of fault-tolerance, some of which is not graceful. Although biological systems survive significant faults, behaviour is commonly degraded or lost. For example in humans or monkeys, the level of behaviour change depends greatly on brain areas that are damaged.

Then, what role does fault tolerance plays in measuring intelligence? From an application point of view, fault tolerance is an important property of designs and system operation. Furthermore, with continuous shrinkage in transistor sizes, the importance of fault-tolerance in highly complex processing system will become increasingly important.

Another more important aspect of fault tolerance requirement is in autonomous system. Here clearly fault tolerance becomes a critical element of endurance and graceful degradation. But is this an important element of intelligence? Although present machine intelligence paradigms only addresses fault tolerance from "an emergent property" perspective, it is possible that fault-tolerance was used a ground-rule for evolutionary development of biological systems, and may lead to yet unknown computational architectures.

Hence, for the short-term, the issue of metrics for fault-tolerance appear to be relevant for autonomous systems in the context of performing tasks in harsh environments and where mechanical and information resource tolerance are important. The tolerance can be graded according to the task and the ability of the machine to complete it in the presence of faults.

8 Consciousness and control

The debate over the neural correlate of consciousness is obviously of most interest to neuromorphic engineering. Our present poor understanding of the underlying neural circuits does not imply that it is not a necessity for autonomous machines. The complex interactions between awareness, planning and survival dictates equipping machines with some level of the "self". The level may be primitive at first. Practical awareness can address sensory representation of the environment and its representations in terms of goals and necessities to survival (e.g. battery charging). The competition of sensory on motor behaviour will need to address priorities and dynamic reward. The representations that emerge from this computation will represent primitive forms of awareness that machines will be capable of processing, but not necessarily of realizing. Realization may emerge as a balanced competition between motor plans, behaviour and reward obtained from behaviour. Hence the development of task-oriented neuromorphic systems will allow the exploration of computational structures and information processing paradigms that can embed such a competition.

In the context of intelligence metrics, the question of consciousness can be stated as that of resource management. The development of a metric framework will have to account for a broad spectrum of sensory, motor and reward situations that could be too complex to represent. One can envision a metric that measures final outcomes based on the essence of task completion measured in terms of energy and survival. That is to be, and to be there in the right time.

9 Complexity, Hierarchy, Regularity and Modularity

The development of design methodologies for highly complex integrated circuits containing tens of million of transistors have taught engineers a number of golden rules in the management of complexity: Hierarchy, regularity and modularity. These human-made engineering rules are similar to the rules that underlie biological systems structures. Representation and learning efficiency (in particular online continuous learning) are the most concerned and affected by the hierarchy, regularity and modularity (HRM) of the underlying structures. The issue of whether HRM issues are relevant to intelligence metrics is similar to those discussed earlier in the context of fault tolerance and representation (e.g. sparseness). HRM of computational structures may affect the optimality of an autonomous system, but may not be critical to its successful operation. Again, the importance of HRM as a ground-rule for autonomy may go beyond optimality and may be critical to the scalability of the architecture and representations. Scalability is relative to the initial conditions and desired bounds. From a practical point of view, it is evident that a HRM-based design will be superior to a design that is flat and that lacks modularity and regularity.

10 Summary and Conclusions

The issues discussed in this position paper converge to the conclusion that in the context of autonomous systems, intelligence metrics should be task oriented and should embed factors such as completion, resources and energy. Completion is easy to assess, with the distinction that it is for practical systems and not for simulations. Resources and energy could be cast to specific implementation, whether software or hardware. Resources will cover aspects of resources used to perform a task, and those available to capture the skills to perform the task. The energy measure will represent the total energy required to perform a task and can easily be measure for software and hardware implementations.

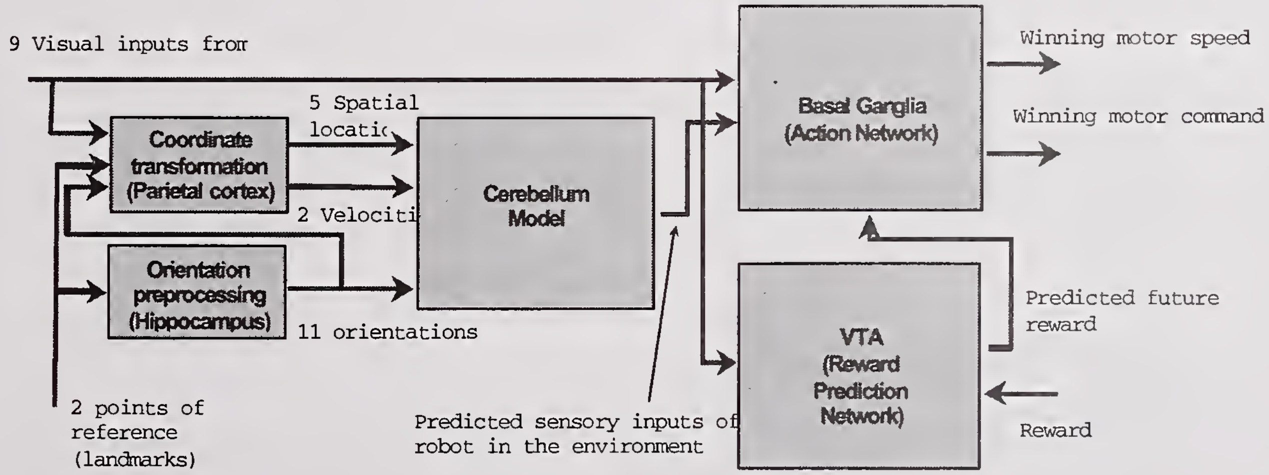

9 Visual inputs from
Coordinate transformation (Parietal cortex)
5 Spatial
locatic
2 Velociti
Orientation preprocessing (Hippocampus)
11 orientations
Cerebellum Model
2 points of reference (landmarks)
Predicted sensory inputs of robot in the environment
Basal Garglia (Action Network)
Winning motor speed
Winning motor command
VTA (Reward Prediction Network)
Predicted future reward
Reward

Figure 1. Sensorimotor system implementing anticipation and reinforcement learning allowing a Khepera robot to track (by rotating in place) a moving target.

11 References

Albus, J. S. (1971). "A theory of cerebellar function." Math. Biosci **10**: 25-61.

Bell, A. J. and T. Sejnowski (1995). "An information-maximisation approach to blind separation and bling deconvolution." Neural Computation 7: 1129-1159.

Coenen, O. J-M. D. (1998). Modeling the Vestibulo-Ocular Reflex and the Cerebellum: Analytical & Computational Approaches. Physics Department, University of California, San Diego.

Coenen, O. J.-M. D. and T. J. Sejnowski (1996). Learning to make predictions in the cerebellum may explain the anticipatory modulation of the vestibulo-ocular reflex (VOR) gain with vergence. Proc. of the 3rd Joint Symposium on Neural Computation, Institute of Neural Computation, University of California, San Diego, and California Institute of Technology.

Etienne-Cummings, R., J. V. d. Spiegel, et al. (2000). "A Foveated Silicon Retina for Two-Dimensional Tracking." IEEE Trans. Circuits and Systems II **47**(6).

Georgopoulos, A. (1995). Motor Cortex and Cognitive Processing. The Cognitive Neurosciences. M. Gazzaniga, MIT Press: 507-518.

Horiuchi, T. and C. Koch (1999). "Analog VLSI-based Modeling of the Primate Oculomotor System." Neural Computation Journal **11**(1): 243-265.

Jabri, M., O. J.-M. D. Coenen, et al. (1997). Sensorimotor integration and control. Extended Abstracts of the NIPS*97 Workshop: Can Artificial Models Compete to Control Robots?, Denver.

Jabri, M. A., R. Coggins, et al. (1996). Adaptive Analog Neural Systems, Chapman and Hall, UK.

Lewis, T., R. Etienne-Cummings, et al. (2000). Towards

Biomorphic Control Using aVLSI CPG Chips. IEEE ICRA, San Francisco, IEEE Press.

Marr, D. (1969). "A theory of cerebellar cortex." J. Physiol **202**: 437-470.

Mead, C. (1989). Analog VLSI and Neural Systems. Reading Massachusetts, Addison-Wesley.

Schaik, A. v. (2000). "An Analog VLSI Model of Periodicity Extraction in the Human Auditory System." *Analog Integrated Circuits and Signal Processing,* Kluwer Academic Publishers **March**.

Sutton, R. S. and A. G. Barto (1981). "Towards a modern theory of adaptive networks: Expectation and prediction." Psy. Review **88**(2): 135-170.

Tilden, M. W. (1994). ""Living Machines"." European Journal of Autonomous Systems.

Grading Intelligence in Machines: Lessons from Animal Intelligence

Subhash Kak
Department of Electrical & Computer Engineering
Louisiana State University
Baton Rouge, LA 70803-5901; kak@ee.lsu.edu

July 20, 2000

In this note I argue that to find a Vector of Intelligence (VI) for a performance metric for machines, it is helpful to look at animal intelligence, which is clearly defined as a spectrum.

All animals are not equally intelligent at all tasks; here intelligence refes to performance of various tasks, and this performance may depend crucially on the animal's normal behavior. It may be argued that all animals are sufficiently intelligent because they survive in their ecological environment. Nevertheless, even in cognitive tasks of the kind normally associated with human intelligence animals may perform adequately. Thus rats might find their way through a maze, or dolphins may be given logical problems to solve, or the problems might involve some kind of generalization. These performances could, in principle, be used to define a gradation.

If we take the question of AI programs, it may be argued that the objectives of each define a specific problem solving ability, and in this sense AI programs constitute elements in a spectrum. But we think that it would be useful if the question of gradation of intelligence were to be addressed in a systematic fashion. The question is best examined in an ecological context; a similar case for an ecological study of machine vision has been made by Gibson.

The issues that we leave out are those related to defining consciousness and quantum approaches to brain processes and intelligence. Although I have personally worked on these issues, I believe they lie outside the scope of the NIST Conference on Performance Metrics for Intelligent Systems.

On Animal Intelligence

According to Descartes, animal behavior is a series of unthinking mechanical responses. Such behavior is an automatic response to stimuli that originate in the animal's internal or external environments. In this view, complex behavior can always be reduced to a configuration of reflexes where thought plays no role. According to Descartes only humans are capable of thought since only they have the capacity to learn language.

Recent investigations of nonhuman animal intelligence not only contradict Cartesian ideas, but also present fascinating riddles. It had long been thought that the cognitive capacities of the humans were to be credited in part to the mediating role of the inner linguistic discourse. Terrace Te85 claims that animals do think but cannot master language, so the question arises as to how thinking can be done without language:

> Recent attempts to teach apes rudimentary grammatical skills have produced negative results. The basic obstacle appears to be at the level of the individual symbol which, for apes, functions only as a demand. Evidence is lacking that apes can use symbols as names, that is, as a means of simply transmitting information. Even though non-human animals lack linguistic competence, much evidence has recently accumulated that a variety of animals can represent particular features of their environment. What then is the non-verbal nature of animal representations?...[For example] learning to produce a particular sequence of four elements (colours), pigeons also acquire knowledge about a relation between non-adjacent elements and about the ordinal position of a particular element. ([6], page 113)

Clearly the performance of animals points to representation of whole patterns that involves discrimination at a variety of levels. But if conceptualization is seen as a result of evolution, it is not necessary that this would have developed in exactly the same manner for all species. Other animals learn concepts nonverbally, so it is hard for humans, as verbal animals, to determine their concepts. It is for this reason that the pigeon has become a favourite with intelligence tests; like humans, it has a highly developed visual system, and we are therefore likely to employ similar cognitive categories. It is to be noted that pigeons and other animals are made to respond in extremely unnatural conditions in Skinner boxes of various kinds. The abilities elicited in research must be taken to be merely suggestive of the intelligence of the animal, and not the limits of it.

In an ingenious series of experiments Herrnstein and Loveland He64 were able to elicit responses about concept learning from pigeons. In another experiment Herrnstein He85 presented 80 photographic slides of natural scenes to pigeons who were accustomed to pecking at a switch for brief access to feed. The scenes were comparable but half contained trees and the rest did not. The tree photographs had full views of single and multiple trees as well as obscure and distant views of a variety of types. The slides were shown in no particular order and the pigeons were rewarded with food if they pecked at the switch in response to a tree slide; otherwise nothing was done. Even before all the slides had been shown the pigeons were able to discriminate between the tree and the non-tree slides. To confirm that this ability, impossible for any machine to match, was not somehow learnt through the long process of evolution and hardwired into the brain of the pigeons, another experiment was designed to check the discriminating ability of pigeons with respect to fish and non-fish scenes and once again the birds had no problem doing so. Over the years it has been shown that pigeons can also distinguish: (1) oak leaves from leaves of other trees, (ii) scenes with or without bodies of water, (iii) pictures showing a particular person from others with no people or different individuals.

Herrnstein He85 summarizes the evidence thus:

> Pigeons and other animals can categorize photographs or drawings as complex as those encountered in ordinary human experience. The fundamental riddle posed by natural categorization is how organisms devoid of language, and presumably also of the associated higher cognitive capacities, can rapidly extract abstract invariances for some (but not all) stimulus classes containing instances so variable that we cannot physically describe either the class rule or the instances, let alone account for the underlying capacity.

Amongst other examples of animal intelligence are mynah birds who can recognize trees or people in pictures, and signal their identification by vocal utterances—words—instead of pecking at buttons Tu82, and a parrot who can answer, vocally, questions about shapes and colors of objects, even those not seen before Pe83.

Another recent summary of this research is that of Wasserman Wa95:

> [Experiments] support the conclusion that conceptualization is not unique to human beings. Neither having a human brain nor being able to use language is therefore a precondition for cognition... Complete understanding of neural activity and function must encompass the marvelous abilities of brains other than our own. If it is the business of brains to think and to learn, it should be the business of behavioral neuroscience to provide a full account of that thinking and learning in all animals—human and nonhuman alike.

Gradation of Intelligence

An extremely important insight from experiments of animal intelligence is that one can attempt to define different gradations of cognitive function. It is obvious that animals are not as intelligent as humans; likewise, certain animals appear to be more intelligent than others. For example, pigeons did poorly at picking a pattern against two other identical ones, as in picking an A against two B's. This is a very simple task for humans. Herrnstein He85 describes how they seemed to do badly at certain tasks:

- Pigeons did not do well at the categorization of certain man-made and three-dimensional objects.

- Pigeons seem to require more information than humans for constructing a three-dimensional image from a plane representation.

- Pigeons seem to have difficulty in dealing with problems involving classes of classes. Thus they do not do very well with the isolation of a relationship among variables, as against a representation of a set of exemplars.

In a later experiment Herrnstein et al. He89 trained pigeons to follow an abstract relational rule by pecking at patterns in which one object was inside, rather than outside of a closed linear figure. Wasserman Wa93,Wa95 devised an experiment to show that pigeons could be induced to amalgamate two basic categories into one broader category not defined by any obvious perceptual features. The birds were trained to sort slides into two arbitrary categories, such as category of cars and people and the category of chairs and flowers. In the second part of this experiment, the pigeons were trained to reassign one of the stimulus classes in each category to a new response key. Next, they were tested to see whether they would generalize the reassignment to the stimulus class withheld during reassignment training. It was found that the average score was 87 percent in the case of stimuli that had been reassigned and 72 percent in the case of stimuli that had not been reassigned. This performance, exceeding the level of chance, indicated that perceptually disparate stimuli had amalgamated into a new category. A similar experiment was performed on preschool children. The children's score was 99 percent for stimuli that had been reassigned and 80 percent for stimuli that had not been reassigned. In other words, the children's performance was roughly comparable to that of pigeons. Clearly, the performance of adult humans at this task will be superior to that of children or pigeons.

Another interesting experiment related to the abstract concept of sameness. Pigeons were trained to distinguish between arrays composed of a single, repeating icon and arrays composed of 16 different icons chosen out of a library of 32 icons Wa95. During training each bird encountered only 16 of the 32 icons; during testing it was presented with arrays made up of the remaining 16 icons. The average score for training stimuli was 83 percent and the average score for testing stimuli was 71 percent. These figures show that an abstract concept not related to the actual associations learnt during training had been internalized by the pigeon. And the performance of the pigeons was clearly much worse than what one would expect from humans.

Animal intelligence experiments suggest that one can speak of different styles of solving AI problems. Are the cognitive capabilities of pigeons limited because their style has fundamental limitations? Can the relatively low scores on the sameness test for pigeons be explained on the basis of wide variability in performance for individual pigeons and the unnatural conditions in which the experiments are performed? Is the cognitive style of all animals similar and the differences in their cognitive capabilities arise from the differences in the sizes of their mental hardware? And since current machines do not, and cannot, use inner representations, is it right to conclude that their performance can never match that of animals?

Another issue is whether one can define a hierarchy of computational tasks that would lead to varying levels of intelligence. These tasks could be the goals defined in a sequence, or perhaps a lattice, that could be set for AI research. If the simplest of these tasks proved intractable for the most powerful of computers then the verdict would be clear that computers are designed based on principles that are deficient compared to the style at the basis of animal intelligence.

Recursive Nature of Animal Behavior

A useful perspective on animal behavior is its recursive nature, or part-whole hierarchy. Considering this from the bottom up, animal societies have been viewed as "superorganisms". For example, the ants in an ant colony may be compared to cells, their castes to tissues and organs, the queen and her drones to the generative system, and the exchange of liquid food amongst the colony members to the circulation of blood and lymph. Furthermore, corresponding to morphogenesis in organisms the ant colony has sociogenesis, which consists of the processes by which the individuals undergo changes in caste and behavior. Such recursion has been viewed all the way up to the earth itself seen as a living entity. Parenthetically, it may be asked whether the earth itself, as a living but unconscious organism, may not be viewed like the unconscious brain. Paralleling this recursion is the individual who can be viewed as a collection of several "agents" where these agents have sub-agents which are the sensory mechanisms and so on.

Logical tasks are easy for machines whereas AI tasks are hard. It might well be that something fundamental will be gained in building machines that have recursively defined behavior in the manner of life. But how such machines could be designed is not at all clear.

A hierarchy of intelligence levels can be useful also in the classification of animal behavior. There does not appear to be any reason that experiments to check for intelligent behavior at different levels could not be devised. Furthermore, experiments could be conducted to determine the difference in ability for individual animals. That such experiments have not been described until now is merely a reflection of the peculiar history of the field.

Concluding Remarks

Study of animal intelligence provides us with new perspectives that are useful in representing the performance of machines. For example, the fact that pigeons learn the concept of sameness shows that this could not be a result of associative response to certain learnt patterns. If evolution has led to the development of specialized cognitive circuits in the brain to perform such processing, then one might wish to endow AI machines with similar circuits. Other questions arise: Is there a set of abstract processors that would explain animal performance? If such a set can be defined, is it unique, or do different animal species represent collections of different kinds of abstract processing that makes each animal come to achieve a unique set of conceptualizations?

Animal behavior ought to be used as a model to define a hierarchy of intelligence tasks. This hierarchy is likely to be multidimensional. Various kinds of intelligence tasks could define benchmark problems that would represent the various gradations of intelligence.

Should VI reflect the degree of recursion in the organization of the intelligence in the machine? Given that the neural organization of the brain consists of "networks of networks", it appears that this be so. On similar grounds, one may assert that the performance of the machine should span several scales. The relative scale invariance of the performance will be a measure of the "quality" of the intelligence.

References

[1] Gibson, J.J. (1979). *The Ecological Approach to Visual Perception.* Houghton Mifflin, Boston.

[2] Herrnstein, R.J., Loveland, D.H. (1964). "Complex visual concept in the pigeon." *Science* 146, 549-551.

[3] Herrnstein, R.J. (1985). "Riddles of natural categorization." *Phil. Trans. R. Soc. Lond.* B 308, 129-144.

[4] Herrnstein, R.J., W. Vaughan, Jr., D.B. Mumford, and S.M. Kosslyn. (1989). "Teaching pigeons an abstract relational rule: insideness." *Perception and Psychophysics* 46, 56-64.

[5] Kak, S. (1996) "Can we define levels of artificial intelligence?" Journal of Intelligent Systems, vol. 6, 133-144.

[6] Terrace, H.S. (1985). "Animal cognition: thinking without language." *Phil. Trans. R. Soc. Lond.* B 308, 113-128.

[7] Wasserman, E.A. (1993). "Comparative cognition: Beginning the second century of the study of animal intelligence." *Psychological Bulletin*, 113, 211-228.

[8] Wasserman, E.A. (1995). "The conceptual abilities of pigeons." *American Scientist*, 83, 246-255.

Biometric Techniques: The Fundamentals of Evaluation

Thomas A. Chmielewski and Paul R. Kalata
Drexel University
Philadelphia, PA 19104
tchmiele@ cbis.ece.drexel.edu, kalata@cbis.ece.drexel.edu

ABSTRACT

While the term biometrics may connote high technology, it simply stands for the concept of recognizing a human being. Today, we use technology to automate the measurements of physical or behavioral characteristic of an individual, so that these measurements may be compared against previously stored data to authenticate an individual's claimed identify. Current biometric systems use diverse measurements, technology and algorithms making it difficult to compare their performance on an equal basis. In general, a candidate biometric system needs to be accessed against the performance requirements of an application. The biometric community uses statistical measures to define the performance of systems. The objective of this paper is to provide a brief tutorial on biometrics and the current measures used to define performance in order to provide information to new biometric users and to stimulate the research community in this rich problem.

Keywords: *acceptance threshold, biometrics, confidence intervals, false accept rate, false reject rate, FAA, FAR, d', performance, pdf, receiver operating curves, ROC*

1. Biometrics: A Brief Tutorial

Humans identify one another by the way faces look and can sometimes identify individuals at a distance based on their stature and gait. Over the years, inventive humans extended their innate identification capability by applying engineering techniques to allow identification of individuals without having the need for someone who explicitly knew the individual. Specifically, the identification problem was solved by relating a physical entity or "secret" information to a person by using:

- something you "**have**" such as a card or key,
- something you "**know**" such as a password or personal identification number (PIN).

Either alone or in combination, possession and knowledge can enable the use of technology to identify a person. ATMs for example require the use of the ATM card (have) plus a PIN (know) to gain access. Since using possession and knowledge for identification purposes cannot distinguish between the correct person and a potential impostor who acquired the possession/knowledge, there is clearly a need for a higher level of positive personal identification.

It is possible to eliminate the aspect of possession and/or knowledge and rely rather on something that the person "**is**", specifically a physiological or behavioral characteristic that can be easily detected, that is time invariant, and that is significantly different across the population of people who will be identified by it. The term *biometric* is used to describe these characteristics which allow identification of an individual. The key advantage of using biometric data to identify a person is that the biometric cannot be stolen, misplaced or forgotten because it is something that the person "**is**", as contrasted to "possession" and/or "knowledge".

Biometric Systems use technology to automate the measurements of physical or behavioral characteristic of an individual, so that these samples may be compared against *previously stored data* to determine if significant similarities exist in order to confirm the samples sufficiently match the stored data hence confirming or denying the individual's identity. In essence, biometric identification is a pattern recognition problem. In order to allow good decisions to be made, we would like maximum variations across individuals, but minimum variation for any given person across time or environmental conditions.

There are two types of problems that a biometric system must handle namely:

- Verification Problem (authentication): confirming or denying a person's claimed identity (Am I who I claim I am ?); this is a one to one matching process.
- Recognition Problem (identification): establishing a person's identity from a set of stored identities; this is a one to many matching process.

Biometrics currently in commercial use for either identification or recognition include: fingerprints, hand geometry, handwritten signatures, voiceprints, face, iris, retinal patterns and thermograms [1]. Certain, physical characteristics such as fingerprints and iris texture, are considered to be "invariant". Behavioral characteristics such as voice and signature. are considered to be "somewhat variable" since they are influenced by physical and emotional conditions and evolve over time.

The retina, the iris and fingerprints are considered truly unique and provide the greatest precision for biometrics [9]. However, other biometrics should not be dismissed, since each biometric provides unique advantages which can be exploited by selecting the correct biometric for the correct application. For instance, INSPASS (Immigration and Naturalization Service Passenger Accelerated Service System) uses a hand geometry system to quickly verify the identity of arriving passengers in speeding up international arrivals at certain North American international airports. People enrolled in INSPASS are given a magstripe card encoded with appropriate data for their hand geometry. Upon arrival, INSPASS travelers swipe their card (**have**), place their hand in a reader (**are**), and then proceed to the customs gate. Coupling the "have" and "are" makes hand geometry a good solution for this application (even though this biometric is not as unique as others) since it is cost effective, readily accepted by most users and exhibits low failure rates in acquiring data.

2. System Functionality

Before a biometric system can operate, a quality sample(s) of the biometric signal, such as a fingerprint, the image of an iris, or speech from users of the system must be obtained. This is called the *enrollment process*. Enrollment usually involves an operator who coaches the users to provide the best biometric input. These inputs are processed and stored as templates or feature vectors that contain the pertinent information used for later biometric data comparison. Additionally, the person's identity in the form of an "ID number" or some data structure is associated with the template. Enrollment should be done under the best of conditions since the quality of data stored during enrollment effects the performance of the system.

Figure 1 illustrates a verification system block diagram.

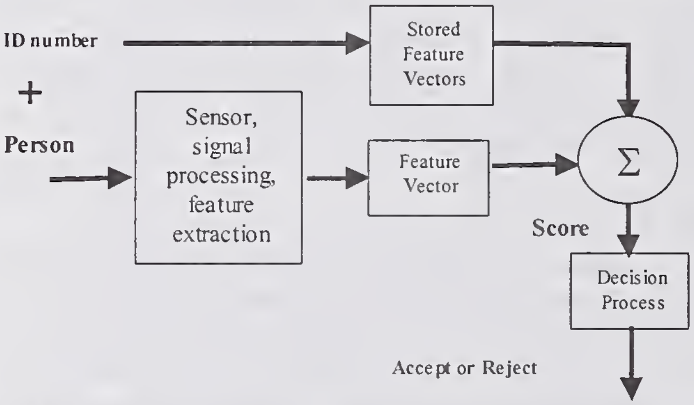


Figure 1. Biometric Verification System Block Diagram

Here it is assumed that a number of individuals have been enrolled in a data base and have been given an " ID number" (such as a bank account). When a person uses the system, their ID number (have) is used to reference a stored feature vector or template in a database. A sensor obtains a biometric sample from the person and then extracts the relevant features from the biometric data into a feature vector. A comparison of the stored feature vector and the computed feature vector is made (*one-to-one process*) generating a matching score. The score is then passed to a decision process. The results of the decision process are acceptance or rejection of the premise that the person at the device is the same as the one who originally generated the feature vector during the enrollment process as referenced by the ID number. Verification acceptance or rejection is based on comparing the matching score to a decision threshold defined by *a priori* statistic of system performance and the application.

The recognition process is significantly different. No ID number is input and the system must compare the feature vector of the person at the device against all stored feature vectors (*a one-to-many process*). If a match is found, then an ID number (and/or other data structure) is retrieved and the person is identified and coupled to this data allowing them access to a building, bank account, etc.

Consider the possible resulting outcomes for a system if a person walks up and attempts to be verified (or identified). There are two possible descriptions of the user: he/she is the correct person (and should be given access) or he/she is an impostor (trying to gain access). Ignoring the case where the biometric system chooses to make no decision, there are two possible outcomes (pass or fail), generating four possible conditions. Table 1 shows these outcomes and conditions: either the correct result occurs, or the system falsely rejects the authentic or falsely accepts the impostor. In the case that the system chooses to make no decision, the user may be given another chance. The performance of a biometric or biometric system is measured by the frequency of false accepts and false rejects.

Table 1. Results of Verification or Identification for a Biometric System User

User	Pass	Fail
Authentic	correct accept- allow access	false reject - refuse access
Impostor	false accept- allow access	Correct reject- refuse access

3. Performance

The quantitative measure of the performance of a biometric verification system is defined by the frequency of false accepts and false rejects [1,12]. These probabilities define how correctly the biometric returns a matching score when a correct individual (authentic) or an incorrect individual (impostor) is presented to the system. While other performance metrics such as the speed of operation, number of templates capable of being stored, and cost are important they are not the focus of this paper.

While we would like to have ideal performance for every biometric (i.e. no False Accepts and no False Rejects), this is unachievable when we consider real world factors such as noise, environmental conditions or the actual discriminating capability of the biometric itself. While not perfect, biometrics are used in many successful applications. Generally, the application dictates the required performance of a biometric. Banks may be willing to accept a certain level of False Accepts but no False Rejects at an ATM in order to keep their customers happy. Alternatively, access to a highly secure facility may not allow any False Accepts but allow False Rejects, since real authentics would be willing to try the multiple times necessary to gain access. A Cost Functional (probability of the decision times a "cost") may also be used to best determine the False Accept versus False Reject trade-off.

3.1 Population Issues: Failure-to-Enroll (FTE)

Even though we would like a biometric to be universally applicable across all users in a given population, there may be some people who cannot use the system due to abnormalities, diseases, injuries, accidents, or degradation of the biometric signal due to their occupation. For instance, masonry workers may wear down their fingerprints so as to make this biometric unreliable. The subset of a population who cannot use a particular biometric are called *outliers*.

Outliers will generally not be able to enroll in a system due to the absence of the biometric or a signal that cannot be converted into a biometric template due to such factors as signal strength, missing characteristics or characteristics present in their biometric not considered by the system. The performance of outliers are categorized by a *failure-to-enroll* (FTE) rate. Numbers to bound the FTE for a biometric can be estimated from the frequency of abnormalities, permanent injuries, or permanent diseases in a given population (or for the world by geographic location) that prevent use of the characteristic. When testing any population, FTE should be accumulated and analyzed.

3.2 System Issues: Failure-to-Acquire (FTA)

Failure-to-Acquire (FTA) is defined as the failure of a biometric system to capture information prior to the extraction of biometric data for the feature vector. This failure may occur during the enrollment process, the verification process or the identification process. It is dependent on the ancillary processes, human factors and external disturbances that may affect the sensor used to acquire the biometric sample. Factors that cause FTAs include: user distraction, or acute injuries or diseases that prevent acquisition of the biometric signal. During testing, FTAs should be accumulated and included in the false reject rate computation (for persons previously enrolled) especially if the entire system performance is being considered rather than just the raw biometrics' performance.

3.3 False Accept Rates and False Reject Rates (FAR, FRR)

Biometric Systems suppliers typically use FAR (False Accept Rate) together with FRR (False Reject Rate) to describe the capabilities of their system (Table 1). FRR is the error rate at which a true authentic (i.e., an individual claiming to be who they actually are) is rejected by the system. FAR is the error rate at which a false authentic (i.e., an individual claiming to be who they are not, i.e., an impostor) is falsely allowed to use the system. FRR and FAR are interrelated by statistics and are dependent on the acceptance or decision threshold of the biometric system under consideration. Being more liberal in the acceptance criterion (a lower threshold), will generally allow more people (both authentics and impostors) into the system. Conversely, being stricter in the acceptance criterion will reject more people (both authentics and impostors). The setting of the threshold is dictated by the requirements of the application.

The relationship between FAR and FRR is easily understood by plotting a distribution of the matching scores of authentics and impostors on the same graph. Sometimes referred to as Authentics-Impostor Distribution Curve or a Performance Histogram, this graph shows the distribution of the population versus a given score of the biometric. The match (or mismatch) score (how well the template matched the biometric sample) is plotted on the horizontal axis and the population frequency on the vertical axis. For statistical analysis, the curves can be normalized so that the area under each curve is one. When normalized in this way the curves become probability distribution functions and may be used to compute probabilities or error rates. Figure 2 shows a typical set of curves. As can be seen the authentics curve (left) and impostor curve (right) overlap. Note that these curves could be interchanged based on the meaning of the horizontal axis. For Figure 2, the better the match between a user and the stored template, the lower the value, with zero (a perfect match) at the far left. The setting of the decision threshold (shown by the vertical bar) defines both the false accept rate (FAR) and false reject rate (FRR). FAR is the area under the impostor curve to the left of the decision threshold. FRR is the area under the authentics curve to the right of the decision threshold. From Figure 2, it is clear that FAR and FRR are interrelated, one cannot specify each value independently!

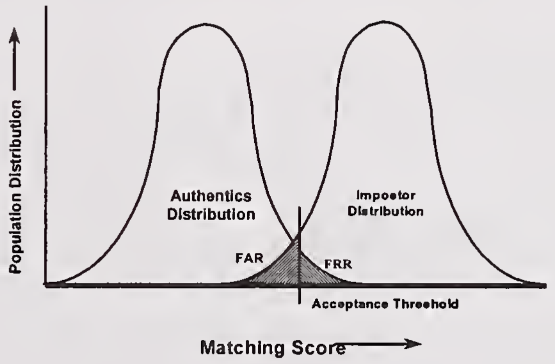


Figure 2. Hypothetical Authentic and Impostor Curves

Impostor and Authentic histograms are different for each biometric. In fact, the actual curves for a given type of biometric may be platform, sensor or algorithm dependent. Obtaining histograms which have sufficient data to perform a reliable curve fit for analysis of their tails requires large amounts of data. The impostor and authentics histograms are usually generated off line by using multiple samples of a closed set of individuals. Hence, for a set of N individuals there will be some number of samples m_N for each individual. By verifying each sample against each other for a given individual an individuals authentics histogram can be generated. The authentics curve for the tested population is obtained by combining all the individual authentics data. Verifying each sample for a given individual against all other individual's samples produces the impostor histogram. Hence, the authentic histogram is a combination ofN people having m_N samples each while the impostor curve is the combination of the verification of each sample for a given person against all other persons' samples.

Performance curves are not necessarily Gaussian. Generally, there is a binning process associated with these performance curves. Binned histograms may show no overlap, incorrectly indicating separation between authentics and impostors. Fitting theoretical curves to the measured data allows computation of FAR/FRR for various thresholds and produces the theoretical tails of the performance curves. Without curve fitting, the granularity of the data bins may not provide sufficient resolution for computation of the FAR/FRR as the decision threshold changes or at extreme points. Fit data may provide more conservative (or more liberal) values than are actually descriptive of the system. It is important to exercise caution with curve fitting since the results may not accurately model the system (especially in the tails) leading to inaccurate probability computations.

Further modifying the shape of performance curves are “goats” and “sheep”. Goats are users who consistently return large distance measures when new samples are compared to their enrolled templates. Sheep, generally a larger part of the population, return small distance measures compared to their enrolled data. The sheep (small variance) and goat (large variance) performance can cause the histogram to exhibit bimodal authentics plots with the goats associated with a smaller secondary mode [1].

3.4 An Empirical Bound on FAR

Consider a biometric that may not be capable of clearly distinguishing characteristics of identical twins (possibly a face biometric). One can put the following bound on the FAR due to identical twins [1]. Statistically, 1 in 80 births are twins and about 1/3 of twins are identical (monozygotic). If we consider 240 births, there are 243 individual and one pair of them are identical twins. The chance of selecting a person at random who has an identical twin is roughly 2/243 = 0.82%. With the assumption that a particular biometric cannot distinguish between identical twins, one can define the minimum False Accept rate at 0.82% due just to the birth of identical twins. Besides user cooperation or technical factors, fraternal twins and parent/offspring may also share the same biometrics value (consider that many fraternal twins still look alike). Putting a number on this contribution to FAR is significantly more difficult.

4.0 The Underlying Probability for Decisions

In biometric identification, a decision to the authentic or impostor must be made based on noisy measurements. To this extent, one must understand the probabilistic nature of the measurement, how it is processed to make a decision and the performance of the decision itself.

Consider a noisy biometric random variable, x, characterized by its probability density function (pdf), $f_X(x)$ with properties [6,7]:

$$f_X(x) \geq 0, \quad \int f_X(x)\, dx = 1.$$

Common pdf shapes include: Gaussian, Uniform, Exponential, Binomial and Poisson. Two important parameters which describe the pdf are the mean and standard deviation (the square root of the variance):

$$\text{mean:} \quad \bar{x} = \int x\, f_X(x)\, dx$$

$$\text{variance:} \quad \sigma_X^2 = \int (x-\bar{x})^2 f_X(x)\, dx.$$

The mean, $\bar{x}$, is "where" the pdf mass is concentrated and the standard deviation, σ_X, is the "spread" of the mass about the mean as illustrated by the triangular pdf in Figure 3.

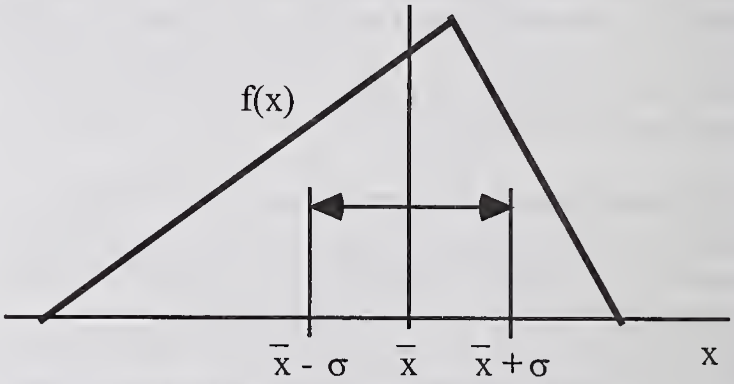


Figure 3. Probability density function

4.1 Biometric Measurement

Consider noisy measurements (or score) of authentic and impostor alternatives $\{m_a, m_i\}$ of $\{x_a, x_i\}$ with zero mean measurement noise $\{e_a, e_i\}$as illustrated by Figure 4.

authentic measurement: $m_a = x_a + e_a$

impostor measurement: $m_i = x_i + e_i$

The variability of the authentic and impostor measurements invariably overlap each other as illustrated by Figure 4.

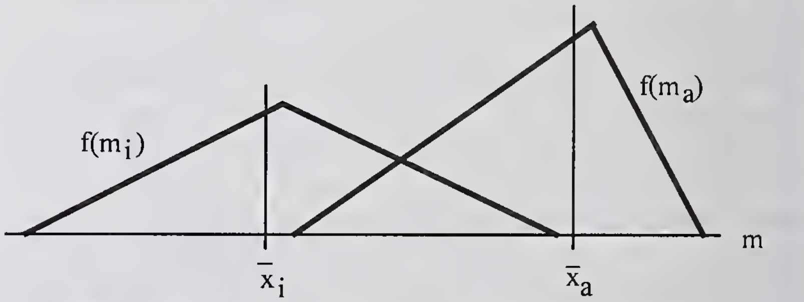


Figure 4. Authentic and impostor measurements

4.2 Biometric Decision

For our analysis, high valued measurements imply an authentic biometric source and low valued measurements imply an impostor source. The decision design [7] is to select a measurement threshold, m_{th}: for measurements exceeding the threshold, decide an authentic source (d_a); alternatively, if the measurement is less than the threshold, decide an impostor source (d_i) as illustrated by Figure 5, i.e.,

$$d_a\text{: if } m \geq m_{th} \quad d_i\text{: if } m < m_{th}$$

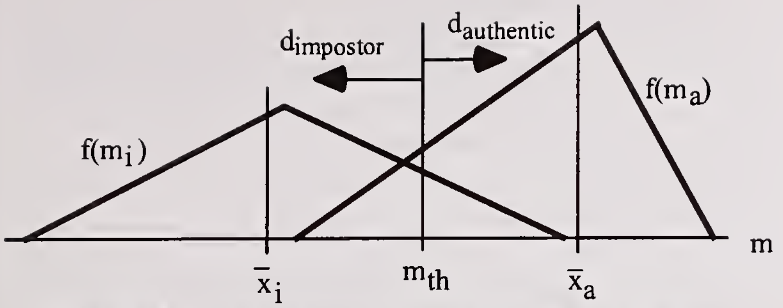


Figure 5. Authentic or impostor threshold decision

For the binary source/decision process there are 4 possible outcomes as illustrated by Table 2 in that:

- an impostor decision with an impostor source results in a correct rejection (dismissal),
- an impostor decision with an authentic source results in a false rejection (mis-detection),
- an authentic decision with an impostor source results in a false acceptance (false alarm), and
- an authentic decision with an authentic source results in a correct acceptance (detection).

The terms {dismissal, mis-detection, false alarm, detection} are common in decision theory and can be interchanged with {correct rejection, false rejection, false acceptance, correct acceptance} respectively as in Table 2.

Table 2. Binary Source/Decision Outcome

Decision	**Source**	
	authentic, x_a	**impostor, x_i**
authentic d_a	correct accept (detect)	false accept (false alarm)
impostor d_i	false reject (mis detect)	correct reject (dismissal)

In virtually all decision cases, no matter where the threshold is set, there will be correct decision and there will be incorrect decisions. The decision design problem is to select the threshold to maximize the (weighted) correct decisions and minimize the (weighted) incorrect decisions.

4.3 Biometric Performance

The decision process performance is determined by the authentic and impostor probabilities which are determined by areas under the pdf's depending. In particular:

Probability of a correct acceptance, $P(d_a|x_a)$, is the area under $f(m_a)$ for $m \geq m_{th}$,

Probability of a false acceptance, $P(d_a|x_i)$, is the area under $f(m_i)$ for $m \geq m_{th}$,

Probability of a correct rejection, $P(d_i|x_i)$, is the area under $f(m_i)$ for $m < m_{th}$, and

Probability of a false rejection, $P(d_i|x_a)$, is the area under $f(m_a)$ for $m < m_{th}$.

Figures 6a, 6b and 6c illustrate these decision probabilities.

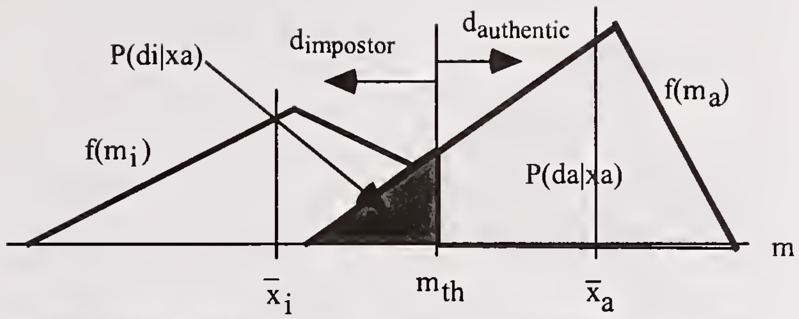


Figure 6a. Probability of Correct Accept and False Reject

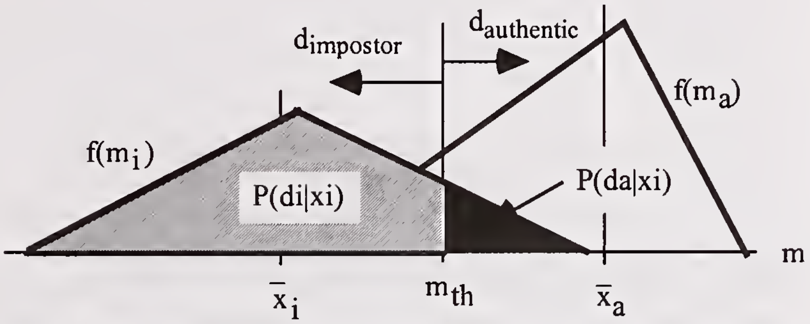


Figure 6b. Probability of Correct Rejection and False Accept

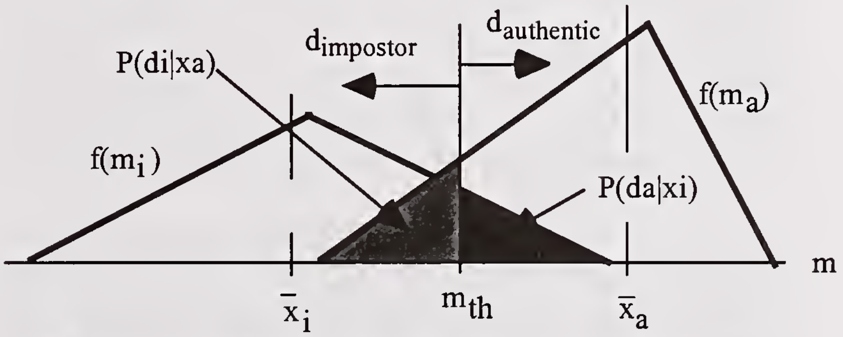


Figure 6c. Probability of False Reject and False Accept

4.4 Cross-Over-Error-Rate (CER)

The value at which the FAR equals the FRR defines the *cross-over-error-rate* (CER) or *equal-error-rate* (EER). The CER may be correlated to the decision threshold that allows this equality. CER provides one method of comparing biometric performance since it is a characteristics of the set of histograms and predefines the threshold setting.

4.5 ROC curves

Another convenient ways to compare the decision process is with "Receiver Operation Characteristic" (ROC) curves which illustrates performance probabilities generated by varying the threshold decision. Figure 7a illustrates the ROC-P(D)/P(FA) performance: Probability of detection (correct acceptance) verses the Probability of false accept. Varying the threshold decision, we can improve in the detection probability, but we also increase the false accept probability. The biometric performance objective is to make this curve as convex to the left as possible and then select which point on this curve is acceptable for biometric identification operation.

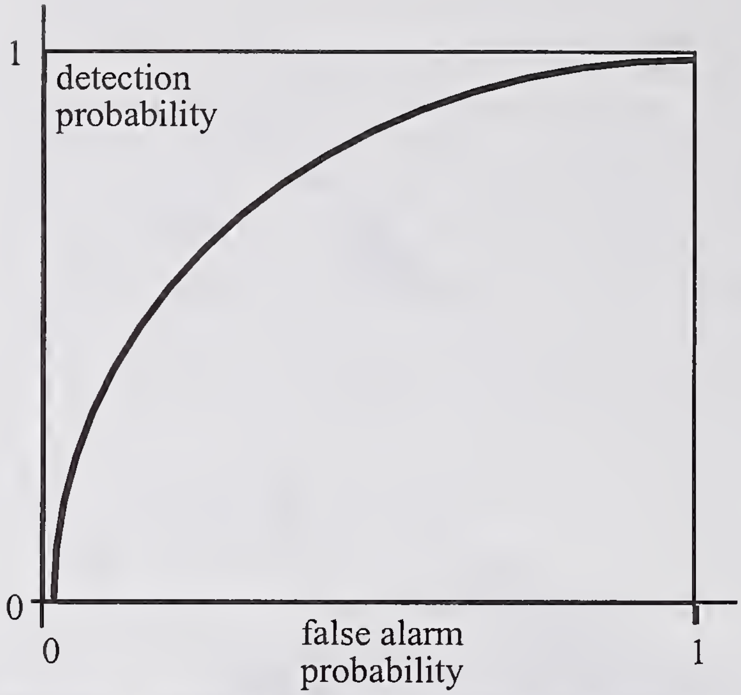


Figure 7a. ROC : P(detection) verses P(false accept)

Another ROC curve is illustrated by Figure 7b, the ROC-P(FR)/P(FA) performance: Probability of false rejects verses the Probability of false accept. This is the ROC curve normally used in biometric literature. Each point on the curve corresponds to a decision threshold and the corresponding FRR and FAR may be easily seen. Varying the threshold decision, we can decrease the false reject probability, but we also increase the false accept probability. The biometric performance objective is to make this curve as concave to the left as possible and then to pick which point on this curve is acceptable for biometric identification operation. As illustrated in Figure 7b the CER (where FAR = FRR) can be easily found from a ROC curve.

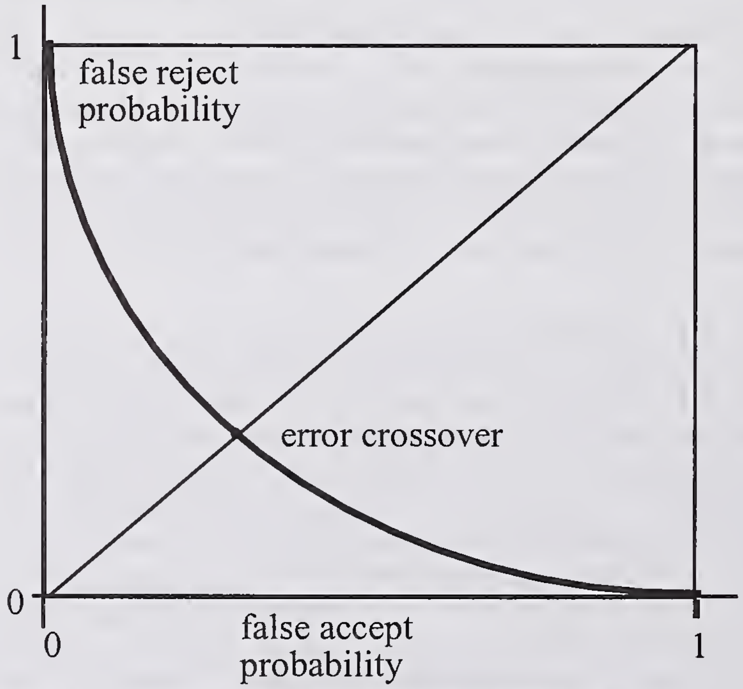


Figure 7b. ROC: P(false reject) verses P(false accept)

4.6 Biometric Confidence

The generation of the biometric pdf's as described above must be made by acquiring test data from the biometric system. Once this is done, the estimates of the means and variance are calculated by the sample means and sample variance:

sample mean: $\bar{x}_s = \frac{1}{N} \sum x_n$

sample variance: $s_x^2 = \frac{1}{N-1} \sum (x_n - \bar{x}_s)^2$

However, these are only estimates of the true mean and variance. Performance as determined by previous sections assumes that we have the true mean and variance. To compensate for only having sample means and sample variances, we have to use "Confidence Levels" and "Confidence Intervals" [8] in describing the biometric performance.

Given N biometric samples, with probability "1-α" confidence level, the sample mean will lie somewhere within a confidence interval between an upper and lower bound which depend on the number of samples, sample mean, the sample variance and a confidence level:

$$\bar{x}_s - \frac{s_x}{\sqrt{N}} t_{\alpha/2,N-1} < \bar{x} < \bar{x}_s + \frac{s_x}{\sqrt{N}} t_{\alpha/2,N-1}$$

where $t_{\alpha/2,N-1}$ is the t-distribution point [8]

$$P(t_{N-1} \ge t_{\alpha/2,N-1}) = \frac{\alpha}{2}$$

Similarly, with probability "1-α" confidence level, the sample variance will lie somewhere within a confidence interval between an upper and lower bound which depend on the number of samples, the sample variance and a confidence level:

$$\frac{(N-1)\, s_x^2}{\chi^2_{\alpha/2;N-1}} < \sigma_x^2 < \frac{(N-1)\, s_x^2}{\chi^2_{1-\alpha/2;N-1}}$$

where $\chi^2_{\alpha/2;N-1}$ is the chi-squared point [8]

$$P(\chi^2_{N-1} \ge \chi^2_{\alpha/2;N-1}) = \frac{\alpha}{2}$$

In biometric system design, we desire to have tight bounds to evaluate/ensure decision performances. Hence, it is obvious from the bound on the sample means, we wish to have a large number of samples to obtain the sample mean with a tight bound. From the "N-1" numerator term in the sample variance bounds, it initially appears that the bound increases with increasing number of samples, but $\chi^2_{\alpha/2;N-1}$ [8] decreases more rapidly than N-1 increases and the net effect is to decrease the sample variance as N-1 increases.

4.7 Decidability Index

The Decidability Index, d' is a measure of the authentic and impostor distributions separation, given by:

$$d' = \frac{\left|\bar{x}_a - \bar{x}_i\right|}{\sqrt{\frac{\sigma_a^2 + \sigma_i^2}{2}}}$$

were $\bar{x}_a$ and $\bar{x}_i$ are the means of the authentic and impostor histograms with variances σ^2_a and σ^2_i. Since the design performance is to maximize the probability of correct decisions and minimize the incorrect decisions, there are two biometric separation properties which can influence the decision performance:

i) maximum separation of the authentic and impostor means, i.e. max: $|\bar{x}_a - \bar{x}_i|$, and

ii) minimize both the authentic and impostor variances, i.e. min: $\{\sigma^2_{x_a}, \sigma^2_{x_i}\}$

If either/both of the above biometric separation properties improve, Figure 8 illustrates the pdf's.

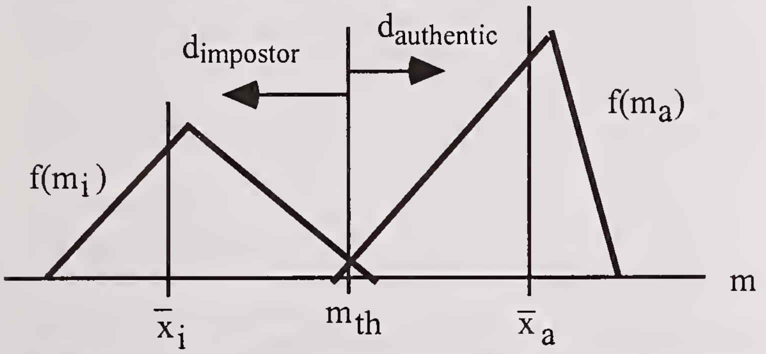


Figure 8. Authentic/Impostor pdf's and threshold decision

Mathematically, d' is independent of the decision threshold and reflects the degree to which any improvement in FAR must be paid for by relaxing of the FRR. As can be seen from the equation the larger d', the better the separation, hence for the best possible discrimination we want a large value of d'. It is obvious that the corresponding Probability of correct/incorrect decisions will improve no matter what the threshold setting is whenever d' is increased.

5. Testing Paradigms

As with using anything new, unfamiliarity can contribute to errors. Biometric systems also exhibit a learning curve. Generally, as users in a test suite become more familiar with the system (habitation) they learn techniques or tricks needed to properly interface to the system and false rejection errors tend to decrease. Proper training can significantly reduce the time to gain system familiarity hence minimizing the initial false reject rate since clues can be given as opposed to being learned by trial and error.

Uncooperative test subjects or unwilling users can skew collected data effecting the measured operative accuracy of a system under test. Thus, it is imperative to design tests that identify or minimize potential human bias during the collection of biometric performance data. Consider the conditions under which data for biometric performance can be collected. Three conditions apply:

- Ideal Data Collection,
- Controlled Real World Data Collection and
- Uncontrolled Real World Data Collection.

In the first case, data is collected and analyzed under the best possible conditions, for instance a user's head may be placed in a chin rest for facial or iris recognition, environmental factors such as ambient temperature and lighting can be controlled. For the second case, data is collected in a real world operational application but under a well controlled experimental situation, for instance having a set of known cooperative subjects use the equipment. The third approach uses data collected at an actual application by actual users, hence it is in an unsupervised, uncontrolled environment.

Each of these scenarios can be used to provide a different perspective on the performance of the system. Ideal data collection allows evaluation of the raw biometric implementation along with its sensors and other components. Controlled real world data introduces the environment as well as an application (but with cooperative users) thereby allowing testing of the biometric system under an actual application. Uncontrolled real world data collection introduces the most variability and is the most difficult. For the last case, it is envisioned that there is an automatic data collection that would report the results of the system's performance over some operational period, there are some issues in this approach as actual data (is it really a false reject or an impostor trying to gain access) would not be known.

5.1 Dual Thresholds

Sometimes "upper and lower bound" acceptance thresholds are sometimes used. This approach allows the user to try a second time if the initial score is between an upper threshold (which would passes the user) and a lower threshold (which would fail the user). Clearly, dual thresholds affects the reported FAR and FRR of the biometric data.

5.2 Attempts and Tries

Collecting statistics for false accepts and false rejects requires a definition of whether the data is collected for a "single try" or is decided after "n tries". The word "try" is used to define a single presentation of the user to the biometric system for measurement (verification). The word "Attempt" defines a cycle of an individual using the biometric system. Most verification devices allow more than one try per attempt and may even internally take multiple samples of the user's biometric for each try. Hence, when collecting and reporting data it is important to define how the counting is done.

Table 3 describes how to count accept and reject data for a "maximum three try" decision process [2]. In this case, accept and reject are defined as whether the user's biometric score is greater or less than a predefined acceptance threshold. If data is collected for multiple users, with multiple attempts per user, the false-reject rate for one-try, two-try or three-try statistics can be computed by using the total of rejects divided by the total attempts, hence for the table shown (assuming each verification results is from a user): FRR = ¾ for one-try statistics, FRR = ½ for two-try statistics and FRR = ¼ for three-try statistics. Other ways of reporting the performance statistics are also possible using this approach for instance one could report the FRR statistics for individual tries.

Table 3 Three Try Decision Counting for a Verification System

Verification Result	One-try Statistics	Two-try Statistics	Three-try Statistics
Accept on 1st try	Accept	Accept	Accept
Accept on 2nd try	Reject	Accept	Accept
Accept on 3rd try	Reject	Reject	Accept
No Accepts in 3 tries	Reject	Reject	Reject

5.3 Some Formal Test Reports

In 1991, Sandia National Laboratories released a report entitled "A Performance Evaluation of Biometric Identification Devices" which describes the testing of a select set of vendors' biometric devices [2]. Specifically, fingerprints, hand geometry, signature, retina and voice biometrics were evaluated and error rate curves generated. A second report titled "Laboratory Evaluation of the IriScan Prototype Biometric Identifier" was issued in 1996 [3]. Both of these reports are a good reference on the procedures and issues that need to be considered for testing of biometric devices. For instance, it is seen that as a user becomes more familiar with a biometric, the false rejection rate decreases, additionally, it was found that retraining and reenrollment of problem users may not always result in higher performance. Sandia's testing attempted to get as many transactions as possible (without too much fatigue or loss of participation from disinterest) from a limited set of users/volunteers.

The FERET (Face-Recognition Technology) program administered by the US Army Research Laboratory [4,5,10] provided a large database of faces collected in a controlled setting (consistent with mug shots or drivers license photos) so that four face recognition algorithms could evaluated under double-blind testing conditions. Results showed dependence on illumination and sensors as well as time between enrollment and verifications (comparison of images taken the same day verses a year apart).

6. Conclusions/Recommendations

Today's automated biometrics technologies are changing due to advances in computer hardware, sensors and algorithms. There are potentially large and diversified uses for biometrics including such applications as: securing Internet credit card based transactions, secure computer logon, ATM access, law enforcement related identification, building access, access to medical records and access to secure facilities. A significant challenge faced in implementing a biometric system is understanding the performance of the system in terms of the tradeoff between FAR and FRR for the particular application. This paper presented an overview of biometrics and statistical measurements currently used to describe the performance of biometric systems. Measuring the performance of a biometric system requires well defined testing and sufficient data to extrapolate performance to the actual user population. Providing quantitative performance to compare different biometrics or even the same biometric using a different platform represents a larger task requiring numerous test subjects, a good definition of the process and a standard data analysis approach. The issue of comparative performance measurements represents a challenge to the research community as well as the biometric industry.

References

1. A. Jain, R. Bolle and S. Pankanti, *Biometrics Personal Identification in Networked Society.* Kluwer Academic Publishers, Boston, July 1999.
2. J. P. Holmes, et al., "A Performance Evaluation of Biometric Identification Devices", *Sandia National Laboratories,* SAND91-0276, June 1991.
3. F. Bouchier, J. Ahrens, G. Wells, "Laboratory Evaluation of the IriScan Prototype Biometirc Identifier", *Sandia National Laboratories*, SAND96-1033, April 1996.
4. P.J. Philips, "FERET (Face-Recognition Technology) Recognition Algorithm Development and Test Results", *Army Research Laboratory*, ARL-TR-995, October 1996.
5. P.J. Philips, et al., "FERET (Face-Recognition Technology) Recognition Algorithms", *Proceedings of the ATRWG Science and Technology Conference*, July 1996.
6. A. Papoulis, *Probability, Random Variables and Stochastic Processes*, 3rd ed, McGraw Hill, New York, 1991.
7. H. Urkowitz, *Signal Theory and Random Processes*, Artech House, 1982.
8. D. Montgomery, *Introduction to Statistical Quality Control*, 3rd ed, John Wiley and Sons, New York, 1996.
9. J.D. Woodward, *Believing In Biometrics*, Information Security Magazine, March 1998.
10. A. Pentland, T. Choudbury, *Face Recogntion for Smart Environments*, IEEE Computer Magazine February 2000, Vol 3, No. 2.
11. M. Negin, T. Chmielewski et al, *An Iris Biometric System for Public and Personal Use*, IEEE Computer Magazine February 2000 Vol 3, No. 2.
12. P. J. Philips, A. Martin et al., *An Introduction to Evaluating Biometric Systems*, IEEE Computer Magazine February 2000 Vol 3, No. 2.

Measuring the quality of visual learning

Giovanni Bianco
Computer Science Serv., University of Verona, Italy
bianco@chiostro.univr.it

ABSTRACT

Biology often offers valuable example of systems both for learning and for controlling motion. Work in robotics has often been inspired by these findings in diverse ways. Nevertheless, the fundamental aspects that involve visual landmark learning has never been approached formally. In this paper we introduce results that explain how the visual learning works. Furthermore, these tools provide bases to measure the quality of visual landmark learning. Basically, the theoretical tools emerge from the navigation vector field produced by the visual navigation strategy. The learning process influence the motion vector field whose features are addressed.

1 INTRODUCTION

Animals are proficient in navigating and diverse methods of biological navigation have been recently studied and categorized as [20]: *guidance, place recognition - triggered response, topological* and *metric* navigations. In order to perform such tasks animals usually deal with identifiable objects in the environment called *landmarks* [21].

The use of landmarks in robotics has been extensively studied [4]. Basically, a landmark needs to possess characteristics such as the *stationarity, reliability* in recognition, and *uniqueness*. These properties must be matched with the nature of a landmark: landmarks can be *artificial* or *natural*. Of course it is much easier to deal with artificial landmarks instead of dealing with natural ones, but the latter are more appealing because their use requires no engineering of the environment. However, a general method of dealing with natural landmarks still remains to be introduced. The main problem lies in the selection of the most suitable landmarks [19].

Recently it has been discovered that wasps and bees perform specific flights during the first journey to a new place to learn color, shape and distance of landmarks. Such flights are termed *Turn Back and Look (TBL)* [11]. Once the place has been recognized using landmarks, insects can then accomplish navigation actions accordingly.

Starting from Biological bases, the system described in this paper selects natural landmarks from the surrounding environment adopting the TBL phase. Once landmarks have been selected suitable navigation movements are computed. Iterating the process of computation of the navigation vector over the whole environment, a vector field is produced.

Studying the navigation vector field two main results are provided:

- the *visual potential function* generating the navigation vector field represents the driving principle to perform visual guidance. When proven to be a *Lyapunov* compliant function, we can state the navigation system exhibits convergence to the goal.
- The *conservativeness* of the navigation vector field provides key information about the quality of landmark learning.

Details about the navigation system and the computation of the potential function can be found in [2] and [3].

This paper addresses the learning process and its organization is as follows. In Section 2 aspects both related to findings on biological learning and to biological navigation will be introduced. In addition, this section addresses former work and research in the field of landmark learning in Robotics. In Section 4 the theoretical principles specifically involved with visual learning are detailed. Final remarks conclude the paper.

2 BIOLOGICAL FOUNDATIONS

Over many decades, studies of the visual performance of bees have exploited the fact that bees keep returning to a profitable feeding site once found, even when it is an artificial food source established by an experimenter.

2.1 Landmark learning

As soon as the bee encounters a novel place, she turns by 180 degrees to inspect the place and performs the initial phase of training, termed the *Turn-Back-and-Look (TBL)* phase [10]. A similar behavior was also observed in other insects thus categorizing this phase a typical behavior of an insect when a new visual learning phase is needed [22, 23].

In references [10, 11] and [14] the details and results on the visual parameters learned by TBL are introduced. Basically, findings show that TBL performed on departure serves primarily for acquiring depth information by exploiting image motion, whereas color, shape and size of landmarks are mainly acquired on arrival.

Attempts to understand in detail the geometric significance of learning flights have only recently been made. Essentially, the flights are invariant in certain dynamic and geometric structures thus allowing the insects to artificially produce visual cues in specific areas of the eyes [24]. Perhaps, the main reason is that the precision for the homing mostly depends upon the proximity of chosen landmarks to the goal [6]. In fact, those flights need to be repeated whenever some changes in the goal position occur [12].

2.2 Landmark guidance

Landmarks guidance in insects is retinoptically driven and animals tend to reduce the discrepancies between the stored view and the actual one by a matching procedure (reviews in [7] and [21]). The survey work presented in [20] addresses biological navigating behaviors from a robotics point of view.

Referring to landmark guidance in bees, the seminal work is presented in [5]. The authors show how bees learn landmarks by storing an unprocessed two dimensional snapshot of the panorama. The model matches landmarks in the stored snapshot with landmarks in the actual image. If this match is performed far from the goal every matched pair could differ both in angular size and compass bearing. These differences drive a bee toward the right position.

3 RELATED WORK

The guidance model introduced in [5] has some shortcomings and interesting extensions have been addressed in recent works. Basically, a guidance strategy that operates with landmarks strives to reduce the differences between the pre-learnt landmarks at the goal position and the same landmarks viewed from a different place. The extraction of landmarks follow different schemas such as in [18] where visual moments are applied on the panorama image to extract prominent features or as in [8] and [16] where unique (small) portions of the whole image, called templates, are extracted.

Operating with landmarks extracted from the panorama, navigation vectors can be computed. Unfortunately, none of the work previously reported tries to handle the mathematical features of the navigation vector fields thus produced.

A formal interpretation of the visual guidance behavior is firstly presented in [1] where two fundamental principles are extracted from the strategy navigation field: the visual potential function and the measure of conservativeness. The latter has been proved to measure the quality of landmark learning whereas the former is a funnel-shaped function that explain why guidance strategies operate with a gradient process to lead the robot to the goal (the global minimum).

4 THE MOTION FIELD

According to what has been previously expressed, starting with local visual information, a vector needs to be computed by the agent which will be used it to perform the next movement. In our case, the computation of the navigation vector is based on information involving the chosen landmarks. How to get navigation information from landmarks is briefly introduced here for completeness and details can be found in [2, 3].

Basically, once landmarks have been learned, they can provide two kind of information to perform motion:

- their actual size, compared to the size learned at the goal site, reports how far/close the agent is to the goal position
- their actual orientation, compared to the orientation learned at the goal site, speaks about the actual left/right shift of the agent.

This kind of data come from each individual landmark and we need to fuse them in order to get the overall navigation vector. Intriguingly, the fusion procedure has strong biological bases as detailed in [20].

To formalize aspects related to the motion field generated in the environment, we call $\mathbf{p}$ the vector representing the robot's Cartesian position $[x\ y]$ in a world reference $\mathbf{W}$. We also define step k the discrete time k of *robot dynamic state.*

Let $\vec{V}(\mathbf{p}(k)) = [V_x(\mathbf{p}(k))\ V_y(\mathbf{p}(k))]$ be the output of the motion strategy at a given step k, i.e. the robot movement at step k. If the robot operates in *position mode*, i.e. at each step it updates its Cartesian position, then

$$\mathbf{p}(k+1) = \mathbf{p}(k) + \vec{V}(\mathbf{p}(k)) \qquad (1)$$

where $\mathbf{p}(k)$ represents the coordinates of robot at step k, and $\mathbf{p}(k+1)$ represents the new position at step $k+1$. The goal position is defined as an equilibrium point $\mathbf{p}^*$ for the system.

The computation over the whole environment of vector $\vec{V}$ defines a vector field $\mathbf{V}$. Let us consider a partial set of equivalent statements about a generic vector field $\mathbf{V}$ [15].

- any oriented simple closed curve $\mathbf{c}$: $\oint_{\mathbf{c}} \mathbf{V} \cdot ds = 0$
- $\mathbf{V}$ is the gradient of some function U: $\mathbf{V} = \nabla U$

The former is related to the concept of *conservativeness* of the field. The latter is concerned with the existence of a *potential function* generating an unique field. From a different point of view, conservativeness is a measure of the quality of landmark learning, whereas the existence of a Lyapunov potential function indicates the robot's capability to reach the goal. The following Section addresses the former aspect. Details of the other aspect can be found in [2, 3].

The robot *Nomad200* was used to accomplish the tests. It includes the *Fujitsu Tracking Card (TRV)* which performs real-time tracking of full color templates at a NTSC frame rate (30Hz).

5 PRINCIPLES FOR LANDMARK LEARNING

A landmark must be reliable for accomplishing a task as detailed in Section 2.1. Landmarks that appear to be appropriate for human beings are not necessarily appropriate for other agents (animals, insects or artificial beings) because of the completely different sensor apparatus and matching systems [19]. Therefore we need to state the meaning of landmark reliability in advance for the system in use before to solve the problem of selecting landmarks.

For our system, a template is a region of the grabbed image identified by two parameters m_x and m_y representing the sizes along X and Y axes. The size ranges from 1 to 8, i.e. from *small* (2^1 pixels wide) to *large* (2^8 pixels wide) templates. The TRV can simultaneously track many templates. For each template the card performs a match in a sub-area of the actual video frame adopting the block matching method [9]. This introduces the concept of

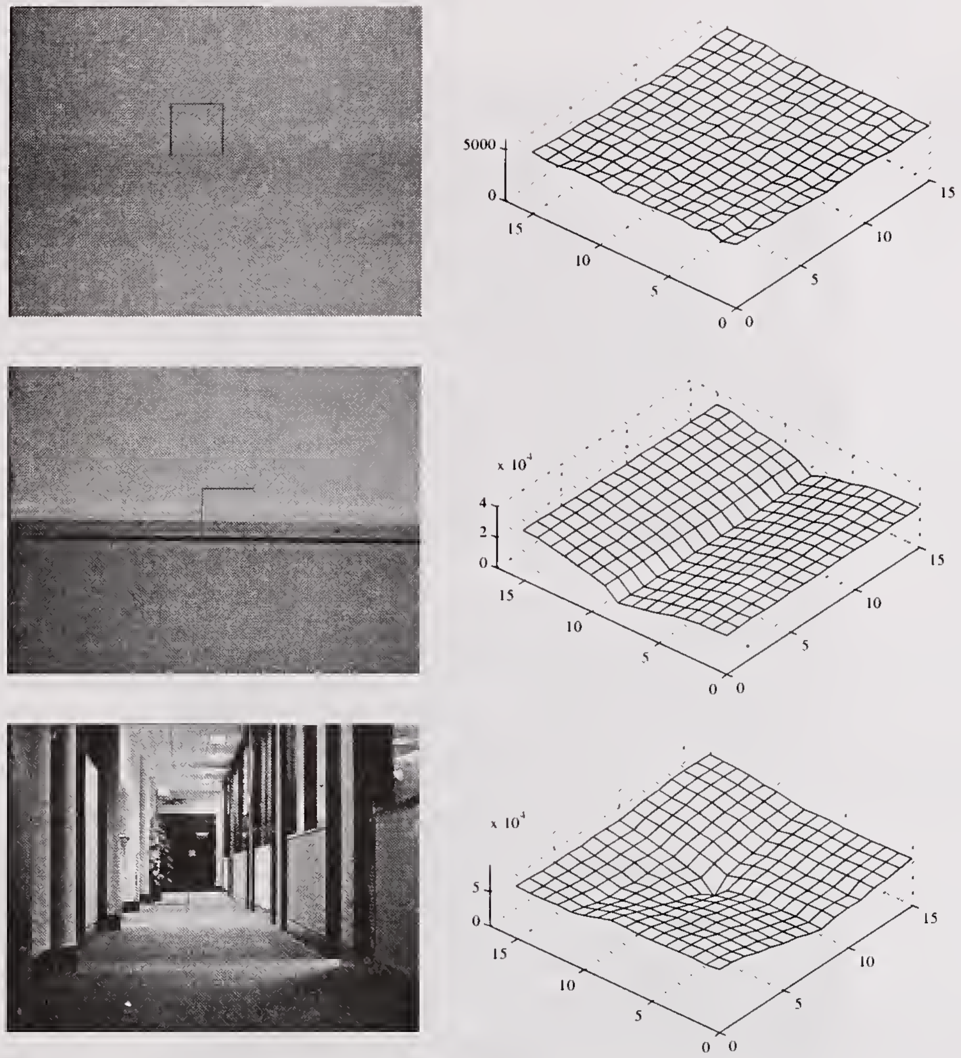


Figure 1: Examples of correlation matrices. These are computed within the local sub-area of the templates (square box in the pictures).

correlation between the template being used and the actual video frame. The sub-area is composed of 16×16 positions in the frame usually taken around the origin (o_x, o_y) of the template (its upper-left corner). The whole set of computed correlation measures forms the *correlation matrix.* Examples of correlation matrices are reported in figure 1.

We can take advantage of the matrices to compute a measure that states upon the reliability of the template under study [17]. As reported in [2, 13] we calculate a figure r, ranging from 0 to 1, which states how deep is the global minimum of the matrix in relation to its neighborhood. Therefore, we define reliable landmarks as *templates which are uniquely identifiable* in their neighborhoods: the greater r the more uniquely identifiable the landmark in its sub-area.

Once that the measure for the reliability of a landmark has been stated, the next step consists of searching the whole panorama for landmarks. There are several degrees of freedom in searching for the best landmarks within a video frame [2], but some simplifications can be introduced: only square templates are used, and the position of a landmark is searched for by maximizing the following:

$$(o_x^*, o_y^*) = \arg \max_{(o_x, o_y) \in \text{grid}} r_l(o_x, o_y) \qquad (2)$$

where $r_l(o_x, o_y)$ identifies the reliability factor for a landmark l whose origin is located in (o_x, o_y) representing a generic place on the grid. The position (o_x^*, o_y^*) represents the cells with the highest r. In order to assure that different landmarks occupy different positions, previously chosen coordinates are not considered. In figure 2, examples of landmarks chosen have been reported. When different sizes are considered, different sets of landmarks are extracted.

The landmarks which have been *statically* chosen are used for navigation tasks. This is done by testing the landmarks to verify that they represent good guides for navi-

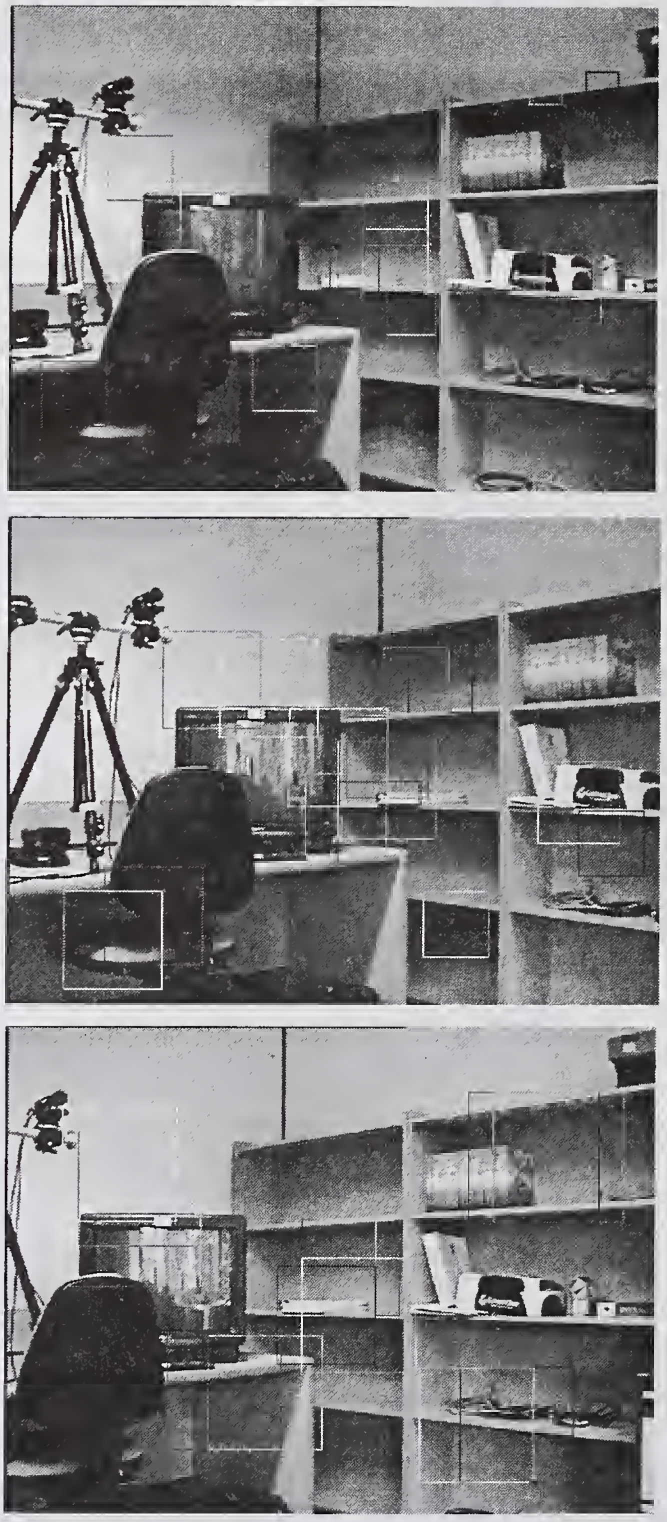


Figure 2: Different choices of landmarks for different landmark sizes. Landmarks are box-shaped.

gation tasks.

TBL helps to verify landmarks by testing whether during the motion the statically chosen landmarks are robustly identifiable. Through a series of stereotyped movements small perturbations (local lighting conditions, changes in camera heading, different perspectives and so on) can influence the reliability of the statically chosen landmarks.

These perturbations to images naturally occur in typical robot journeys thus allowing us to state that the TBL phase represents a *testing framework* for landmarks. In other words, the robot tries to *learn* which landmarks are suitable for use in real navigation tasks by simulating the conditions the robot will encounter along the paths. At the end of the TBL process only those landmarks whose reliability r_l is above a certain threshold ϵ are suitable to be used in navigation tasks.

The reliability factor r_l for landmark l is continuously computed during the TBL phase through the following:

$$r_l = \frac{\sum_{i=1}^{TBL} r_l^i}{TBL} \tag{3}$$

where TBL is the total number of steps exploited till that time, and r_l^i is the reliability of landmark l calculated at time i. In the tests, at the end of the phase, TBL usually consists of 400 steps (it takes about 13 seconds to be performed). The set of landmarks is tracked along the whole TBL phase and r_l is continuously monitored for each landmark (details in [13]).

5.1 *The quality of learning*

There are strong connections between the learning phase and navigation actions. The conservativeness of the motion field bridges these two aspects.

A vector field $\mathbf{V}$ is said to be *conservative* when the integral computed on any closed path is zero. Conversely, if the field is not conservative then diverse potential functions can be associated with the field. This translates into *non-repeatability* of robot navigation trails in [13].

If the vector field is defined on a connected set in the environments, then the null *circuitation* property is equivalent to [15]:

$$\frac{\partial V_x(x,y)}{\partial y} = \frac{\partial V_y(x,y)}{\partial x} \tag{4}$$

We can measure how this equation differs from the theoretical null value as follows:

$$\frac{\partial V_x(x,y)}{\partial y} - \frac{\partial V_y(x,y)}{\partial x} \tag{5}$$

The property expressed by Equation 5 is referred to as *degree of conservativeness*. The degree of conservativeness of the vector field computed with a threshold set to 0 and landmarks sized 6 is shown in figure 3. Only small regions of the whole area roughly satisfy the constraint.

A small change in the threshold for TBL can dramatically change the situation. In figure 4 the degree of conservativeness for each point is plotted.

A key consideration is concerned with the scale along Z: it is about one order of magnitude less than the one reported in figure 3. A trend toward a conservative field is thus becoming evident.

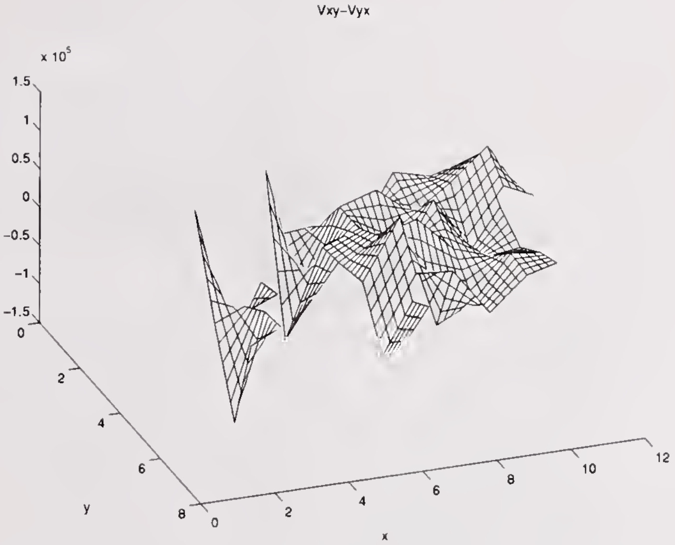


Figure 3: Conservativeness of a vector field computed with a TBL threshold of 0 and landmarks sized 6

The situation obtained with a threshold of TBL set to 0.2 has been reported in figure 5. A large area of the environment has a degree of conservativeness that roughly equals 0.

Similar considerations can be expressed dealing with a different landmark size [1]. The *template* of the graph is the same as before. Therefore, with a good choice of threshold the field becomes conservative regardless of the size of the landmarks.

6 CONCLUSIONS

Landmarks learning for robots can take inspiration from Biology but it needs to be well formalized for its efficient implementation in artificial agents. First, a definition for landmark reliability must be stated. Second, a measure that can assess about the quality of the learning phase needs to be introduced.

In this paper, we have shown how both these aspects can be efficiently addressed. Particularly, we have shown how the learning phase affects the navigation motion field. Further improvements to this study can be achieved by the use of omni directional visual sensors.

References

[1] G. Bianco. *Biologically-inspired visual landmark learning and navigation for mobile robots.* PhD thesis, Department of Engineering for Automation, University of Brescia (Italy), December 1998.

[2] G. Bianco and A. Zelinsky. Biologically-inspired visual landmark navigation for mobile robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Kyongju (Korea), October 17-21 1999.

[3] G. Bianco and A. Zelinsky. Dealing with robustness in mobile robot guidance while operating with visual

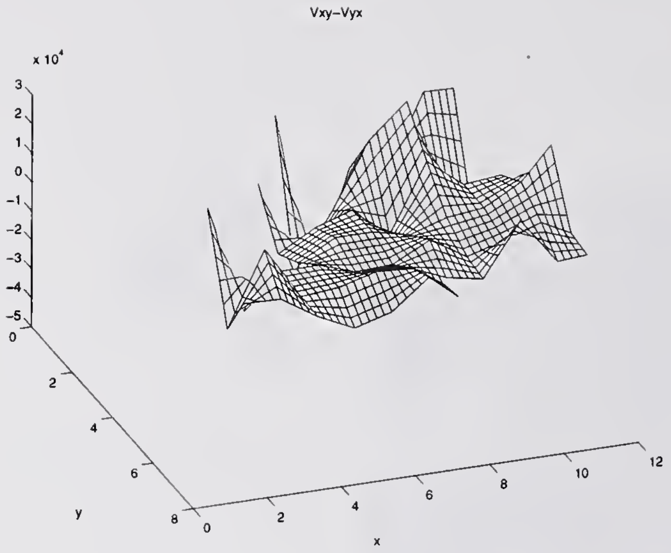


Figure 4: Conservativeness of a vector field computed with a TBL threshold of 0.1 and landmarks sized 6

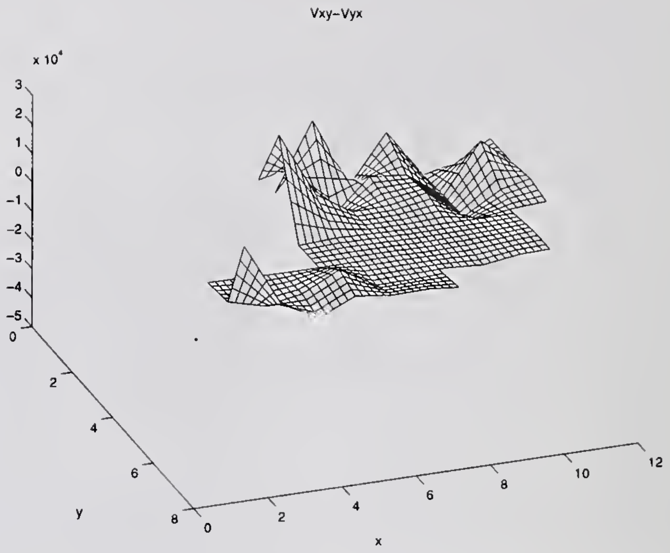


Figure 5: Conservativeness of a vector field computed with a TBL threshold of 0.2 and landmarks sized 6

strategies. In *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco (CA), April 24-28 2000.

[4] J. Borenstein, H. Everett, and L. Feng. *Where am I? Sensors and Methods for Mobile Robot Positioning.* The University of Michigan, April 1996.

[5] B. Cartwright and T. Collett. Landmark learning in bees. *Journal of Comparative Physiology*, A(151):521–543, 1983.

[6] K. Cheng, T. Collett, A. Pickhard, and R. Wehner. The use of visual landmarks by honeybees: Bees weight landmarks according to their distance from the goal. *Journal of Comparative Physiology*, A(161):469–475, 1987.

[7] T. Collett. Landmark learning and guidance in insects. *Phil. Trans. R. Soc. London*, B(337):295–303, 1992.

[8] I. Horswill. *Polly: a Vision-Based Artificial Agent.* PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, May 1993.

[9] H. Inoue, T. Tachikawa, and M. Inaba. Robot Vision System with a Correlation Chip for Real-time Tracking, Optical Flow and Depth Map Generation. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1621–1626, Nice, France, May 12–14 1992.

[10] M. Lehrer. Bees which turn back and look. *Naturwis*, (78):274–276, 1991.

[11] M. Lehrer. Why do bees turn back and look? *Journal of Comparative Physiology*, A(172):549–563, 1993.

[12] M. Lehrer. Honeybees' visual spatial orientation at the feeding site. In M. Lehrer, editor, *Orientation and communications in arthropods*, pages 115–144. Birkhauser verlag, Basel/Switzerland, 1997.

[13] M. Lehrer and G. Bianco. The turn-back-and-look behaviour: Bee versus robot. *Biological Cybernetics*, 2000.

[14] M. Lehrer and T. Collett. Approaching and departing bees learn different cues to the distance of a landmark. *Journal of Comparative Physiology*, A(175):171–177, 1994.

[15] J. Mardsen and A. Tromba. *Vector calculus.* W.H. Freeman and Company, 1996.

[16] Y. Matsumoto. *View-Based Approach to Mobile Robot Navigation.* PhD thesis, Graduate School of Engineering, The University of Tokyo (Hongo 7-3-1, Bunkyo-Ku), February 1998.

[17] T. Mori, Y. Matsumoto, T. Shibata, M. Inaba, and H. Inoue. Trackable attention point generation based on classification of correlation value distribution. In *JSME Annual Conference on Robotics and Mechatronics (ROBOMEC 95)*, pages 1076–1079, Kavasaki (Japan), 1995.

[18] J. Salas and J. Gordillo. Robot location using vision to recognize artificial landmarks. In *SPIE*, volume 2354, pages 170–180, 1995.

[19] S. Thrun. A bayesian approach to landmark discovery and active perception in mobile robot navigation. Technical report, School of Computer Science Carnegie Mellon University, 1996.

[20] O. Trullier, S. Wiener, A. Berthoz, and J. Meyer. Biologically based artificial navigation systems: Review and prospects. *Progress in Neurobiology*, 51:483–544, 1997.

[21] R. Wehner. Arthropods. In F. Papi, editor, *Animal Homing*, pages 45–144. Chapman and Hall, London, 1992.

[22] J. Zeil. Orientation flights of solitary wasps 1: Description of flights. *Journal of Comparative Physiology*, A(172):189–205, 1993.

[23] J. Zeil. Orientation flights of solitary wasps 2: similarities between orientation and return flights and the use of motion parallax. *Journal of Comparative Physiology*, A(172):207–222, 1993.

[24] J. Zeil, A. Kelber, and R. Voss. Structure and function of learning flights in bees and wasps. *Journal of Experimental Biology*, 199:245–252, 1996.

Autonomous Mental Development and Performance Metrics for Intelligent Systems

Juyang Weng
Department of Computer Science and Engineering
Michigan State University
East Lansing, MI 48824
http://www.cse.msu.edu/~weng/

Abstract

In this paper, some resent advances in neuroscience, psychology, robotics and machine intelligence are briefly reviewed. They prompt us to pay attention to the fundamental difference between the way human intelligence is developed and the traditional engineering paradigm for developing a machine. They make us rethink the issue of intelligence. This position paper proposes that a fundamental criterion for a true intelligent system is not really what it can do in a special setting, but rather, its capability for autonomously and incrementally developing its cognitive and behavioral capability through online real-time interactions with its environment, directly using its sensors and effects, a process called mental development in neuroscience and psychology. The term ``mental'' here includes cognitive, behavioral, sensorimotor and other mental skills that are exhibited by animals and humans. The new direction of autonomous mental development for machines will create a new kind of machines, called developmental robots. With new perspectives from developmental robots, the performance metrics for machine intelligence will undergo a revolution. They will fundamentally change the current fragmented landscape of the AI field by shifting the emphasis of measuring ad hoc capability of performing a task-specific application to a systematic measurement of mental developmental capabilities. Such performance metrics can be adapted from those for humans — a series of tests well developed by a well-established field called psychometrics.

1 Background

Human understanding of the ways our own minds work, the power and limitation of existing machines, as well as the relationship between humans and machines have greatly improved over the last 50 years. It is now clear that a *developed* human mind, that of a normal human adult, is extremely complex. It is also clear that the early optimism in the 60's and the 70's about a quick progress in artificial intelligence such as vision, speech, and language, was not well founded, at least not so with the traditional approaches that have been extensively experimented with so far. However, the past work with the traditional approaches is by no means unimportant. In fact, they are the womb and incubator for the birth and growth of a drastically different approach — autonomous mental development. This new direction is expected to become a revolution in the course of machine intelligence[1]. As Thomas S. Kuhn wrote in his book titled *The Structure of Scientific Revolution* [1]: "Because it demands large-scale paradigm destruction and major shifts in the problems and techniques of normal science, the emergence of new theories is generally preceded by a period of pronounced professional insecurity. As one might expect, that insecurity is generated by the persistent failure of the puzzles of normal science to come out as they should. Failure of existing rules is the prelude to a search for new ones."

The puzzle pieces from recent advances in related fields start to reveal a picture of *mental development,* which is no longer a total myth that is beyond human comprehension, but can be explained in terms of computation. In the following we briefly summarize these new thought-provoking advances.

[1] A more detailed discussion on this issue is available in the proceedings of Workshop on Development and Learning, funded by NSF and DARPA, held at Michigan State University, East Lansing, MI, April 5 – 7, 2000 (http://www.cse.msu.edu/dl/). This workshop was attended by about 30 distinguished researchers in neuroscience, developmental psychology, machine intelligence and robotics who are working on related subjects in their fields. The goal of this workshop was to discuss the state-of-the-art in research on mental development and to discuss, initiate and plan future research on this subject.

1.1 Neuroscience and psychology

A traditional view is that human brain is very much pre-determined by human genes. With this view, the brain unfolds its pre-determined structure during the development, which starts from the time of conception. This structure serves as a placeholder of information that is acquired from the environment. However, recent advances in brain plasticity have begun to reveal a very different picture of brain development. For example, researchers at MIT [2] have discovered that if the optical nerves from thc eyes is rewired into the auditory cortex of the primate (ferret) early in life, the primate's auditory cortex gradually takes on representation that is observed in normal visual cortex. Further, the primates have successfully learned to perform vision tasks using the auditory cortex. In other words, the rewired ferrets can see in the sound zone. This discovery seems to suggest that the cortex is governed by self-organizing mechanisms, which derive representation and architecture according to the input signals, either visual or auditory. As another example, studies by researchers at the University of California at San Francisco [3] showed that the finger skin areas from which a neuron in somatic cortex receives sensory signals (called receptive field of the neuron) can change according to sensory experience. If multiple fingers of the adult monkeys receive consistent synchronized pulse stimuli from a cross-finger bar for several days, the receptive field changes drastically, from covering only a single finger in normal cases to covering multiple fingers. This result appears to indicate that the self-organizing program of our brain autonomously selects the source of sensory input within a candidate area according to the statistical properties of the actual sensory signal that is received. These and other related studies on the brain plasticity prompt us to rethink the traditional rigid view about the brain. It appears that the developmental program of the brain does not rigidly determine the brain's architecture and representation. For example, it might determine what statistical properties of the sensory signals should be used and how these properties are used to derive the representation and architecture of the brain.

In recent years, computational modeling of neural development has become a very active subject of study in neural science and psychology. For example, there have been several computational models for the development of response patterns in the retina, the lateral geniculate nucleus, and simple cells in the visual cortex. A subject that is now very actively studied is the mechanisms for developing orientational selectivity in the simple cells of the visual cortex. Although most computational models of developmental mechanisms have been concentrating on early processing (early in the order of processing steps in the brain), such a trend will certainly extend to later processing when global developmental models are increasingly studied for robots. Psychology has begun to move from qualitative descriptive models to more rigorous quantitative models for studying cognitive and behavioral processes. Some recent works in psychology has started to explain the global process of mental development using the computational element of networks [4]. Another new trend in psychology is to use explicit dynamics models to explain some well-known developmental facts about infant behaviors (e.g., the work at Indiana University [5]). These quantitative studies have begun to produce results that are more clearly understandable and verifiable than vague verbal theories and arguments.

1.2 Robotics and Machine Intelligence

Although autonomous mental development in humans is a well-known fact, the counterpart for machines did not receive serious attention until middle 90's. It has long been believed that the approach to machine intelligence does not have to follow what human minds do, just like modern airplanes which do not fly like birds. Gradually, many AI researchers started to realize that machine intelligence requires much more cognitive and behavioral capabilities than most had realized. Flying is a very simple problem in comparison with machine intelligence. Further, many AI researchers have already realized that machine intelligence requires "grounding" — concepts must be grounded on real sensory experience about the physical world, which in turn requires the machine to have a sensor-rich body (i.e., embodiment) that can directly sense stimuli from the physical world and act upon what it senses. However, grounded sensing and action, including learning, has been extensively studied and experimented with in robotics for many years. Why then does the reality of intelligent machines seem so remote? Since 1996, I argued [6] that what has been sorely missing from machines is *autonomous mental* *development*, or simply called *mental development*.

Autonomous mental development requires a true revolution in the way engineering has been done (i.e., paradigm) for thousands of years. The current *manual developmental paradigm* is as follows:

1. *Start with a task:* Given a task to be executed by a machine, it is the human engineer who understands the task (not the machine).
2. *Design task-specific representation:* The human engineer translates his understanding into a representation (e.g., giving some symbols or rules that represent particular concepts for the tasks and the correspondence between the symbols and physical concepts). The representation reflects how the human engineer understands the task.
3. *Task-specific programming:* The human engineer then writes a program (or designs a mechanism) that controls the machine to perform the task using the representation.
4. *Run the program on the machine.* Sensory data may be used to modify the parameters of the task-specific representation. However, since the program is of special purpose for the task, the machine does not even know what it is doing at all. All it does is running the program.

The new paradigm, *autonomous developmental paradigm,* for constructing developmental machines or robots, is as follows:

1. *Design body:* According to the general ecological condition in which the robot will work (e.g., on-land or underwater), human designers determine the sensors, the effectors and the computational resources that the robot needs and the human designs a sensor-rich robot body.
2. *Design developmental program:* Human designer designs the *developmental program* for the robot and starts to run this program.
3. *Birth:* The human operator loads the developmental program onto the computer in the robot body.
4. *Develop mind:* Humans mentally "raise" the developmental robot by interacting with it. The robot develops its mental skills through real-time, online interactions with the environment, including humans (e.g., let them attend special lessons). Human operators teach robots through verbal, gestural or written commands very much like the way parents teach their children. New skills and concepts are learned by the robots daily. The software (brain) can be downloaded from the robots of different mental ages to be run by millions of other computers, e.g., desktop computers.

Mental development has long been mistakenly thought of as being simulated by traditional machine learning techniques (e.g., neural network techniques). In fact, all the traditional machine learning uses the *manual* developmental mode but mental development requires the *autonomous* developmental mode. What is the basic difference? *With autonomous mental development, machines will be able to learn subjects that their programmers do not know, or have not even thought about, just like human children who can learn subjects that their parents do not know.* The essence of autonomous mental development by machines is the capability of learning directly, interactively, and incrementally from the environment using the learner's own sensors and effectors. Therefore, a computer that has only impoverished sensors and effectors cannot do mental development well. A neural network that can only accept human edited offline sensory data does not develop its mind either, even if it can learn incrementally. A *developmental robot* is a robot that runs a developmental program and is allowed to learn and practice autonomously in the real physical world.

Although the concept of developmental program for machines is very new, a very rich set of techniques useful for developmental programs have already been developed in the past several decades in the fields of pattern recognition, robotics and machine intelligence, especially techniques applicable to high-dimensional data. These new techniques are being used in very innovative ways for developmental programs. Several developmental programs have been designed and tested on robots. Running a developmental program, the robots interact with the environment in real time using their sensors and effectors. Internal representation, perceptual capabilities and behavioral capabilities are developed autonomously as a result of interaction of the developmental program with the environment. Humans interact with such robots only through the robot's sensors, as a part of the environment. Just like the nature-nurture interaction for human mental development, the cognitive and behavioral skills of such a robot result from extensive interaction between what is programmed ("innate" developmental program) and what is sensed through real-time online experience. The mind and intelligence emerges gradually from such interactions.

Early examples of such developmental robots include Darwin V at The Neurosciences Institute, San Diego and the SAIL at Michigan State University, developed independently around the same time with very different goals. The goal of Darwin V [7] was to provide a concrete example for how the properties of more complex and realistic neural circuits are determined by the behavioral and environmental interactions of an autonomous device. Darwin V has been tested for the development of generalization behaviors in response to visual stimuli at different positions and orientations (visual invariance learning). It has also been tested for the association of aversive and appetitive stimuli with visual stimuli (value learning). SAIL was designed as an engineering testbed for developmental programs that are meant for scaling up for complex cognitive and behavioral capabilities [8]. SAIL-2 developmental program has been tested for automatic derivation of representation and architecture through development of association between visual stimuli of objects and eye aiming for the objects (object evoked visual attention), between visual stimuli of objects and arm pre-reaching for the object (vision evoked object reaching), between voice stimuli and arm actions (verbal command learning and execution) and between visual stimuli and locomotion effectors (vision-guided navigation). Other studies for online learning directly from sensors are in the direction towards fully autonomous developmental systems. The work at MIT associates video images of objects with synchronized voices (pronounced verbal name of the object) [9]. The work at the University of Massachusetts at Amherst investigated the use of coupling of robot leg joints that have been observed in infants to reduce the search space for a desirable turning gait [10]. Although the history of developmental robots is very short, some experiments by the above studies have demonstrated capabilities that have never been achieved by the traditional methods, such as in visual recognition, verbal communication, hand-eye coordination, autonomous navigation, value acquisition (learning the value of actions), and multimodal association in real time. We are aware that more groups in the US and other countries have already started to investigate this new direction.

2. Some Major Characteristics of Research on Mental Development

2.1 More tractable

It is known that a developed adult human brain is extremely complex, as an epigenetic product of long-term and extensive interactions with the complex human world. The developmental principles for the brain in the complex human world, however, should not be as complex as the human world itself. For example, the visual world is very complex, but the developmental principles that are used by the brain to derive various filters for processing visual signals should not be as complex as the visual world itself. Therefore, computational study of cognitive development could be more tractable than traditional approaches to understanding intelligence and constructing intelligence machines.

2.2 Unified framework

Studies of cognitive development will establish a unified framework for our understanding of a wide variety of cognitive and behavioral capabilities. Discovery of mechanisms responsible for developing cognitive and behavioral capabilities in humans requires more systematic work than an account of a particular individual capability, such as visual recognition in a simplified setting alone or stereotyped walking alone. Sharing of common developmental principles by visual and auditory sensing modalities, as revealed by recent neuroscience studies, will encourage scientists to discover further underlying developmental principles that are shared not only by different sensing and effector modalities, but also by different higher brain functions.

Traditionally, vision and speech have been considered very different, both for humans and for machines. For the same reason, traditional methods for different AI problems are typically very different, resulting in what is well known now as the fragmentation of the AI field. Potentially, AI can be applied to all possible areas of human life and each application area potentially can lead to a fragment of AI if it is treated in an *ad hoc* way. The unified framework of cognitive development will fundamentally change the current fragmented landscape of AI in the years to come, since different applications correspond to different lessons that can be taught to the same developmental robot at different mental ages. We will also see much more interactions and collaborations among scientists and engineers in neuroscience, psychology, robotics, artificial intelligence and other related fields, due to the very similar research issues these fields face under the theme of autonomous cognitive development.

2.3 Task-nonspecific

In contrast to the task-specific nature of the traditional engineering paradigm in AI, developmental programs for machines will be *task-nonspecific*. The power of a developmental program is its general applicability to many different tasks. A developmental program may contain certain pre-processing stages that are specific to some type of sensors or effectors, such as camera or touch sensors. In this sense, it is body-specific (or species-specific). However, it is not task-specific. A developmental program can be run to develop skills for many different tasks, with simpler skills being learned to prepare skills for learning more complex skills. Recently, the scientific community has gained a more complete understanding of human intelligence. As Howard Gardner put it in his book *Multiple Intelligences* [11], human intelligence is multiple, including linguistic, logical-mathematical, musical, bodily-kinesthetic, spatial, interpersonal, and intrapersonal intelligences. This is a rough classification of a very rich ensemble of inter-related cognitive and behavioral capabilities that give rise to human intelligence. The same is true for machine intelligence. Any particular capability that we regard as intelligence in a general setting, such as the visual capability of recognizing various persons on a busy street or the language capability of talking about technology, is not an isolated single thing. It requires the support of many skills developed through extensive real-world experience via sensors and effectors.

2.4 Computational

Further, developmental mechanisms seem to be very much quantitative in nature and thus require clear *computational models*. We will see more complete computational models for mental development that can be simulated on computers and robots for many different environmental conditions and the results can be verified against studies about humans. We will see more efforts on computational modeling of mental development, for humans and machines, that are clearly understandable, implementable on machines and can be subject to rigorous verification and comparison. This will indicate the maturation of the related fields.

2.5 Recursive and active

Development discourages any static or rigid view of the mind. A developed human mind is a snapshot of many years of *recursive and active* mental construction by the developmental program in the human genes, utilizing the sensory and action experience through life time. The term *recursive* means that later mental development relies on the cognitive and behavioral capabilities that have been developed earlier. The term active means that each individual plays an active role in the development of his or her mind --- different actions lead to different experience. The same is true for developmental robots. The recursive and active nature of development discourages the approach of collecting offline data and spoon-feeding them into a machine, which is a prevailing approach in current machine learning studies. Sensory data cannot be pre-specified since what sensory data is sensed depends on online action executed in real time.

2.6 Developmental capabilities as unified metrics for machine intelligence

The *criteria for measuring machine intelligence* will fundamentally change. The metrics that can be used to measure the power of such a new kind of machine is primarily their autonomous interactive learning capabilities in complex human environments. In other words, it is the capability of mental development instead of what the machine can do under a pre-specified setting. Such performance metrics can be adapted from those used by clinical psychologists for testing the cognitive development of human infants (e.g., The Bayley Scales of Infant Development) and children (e.g., The Leither International Performance Scale). The mental age that is used for measuring human intelligence in these tests will be adapted to a scale for measuring machine intelligence. This is a fundamental change from the current metrics that measure what a machine can do under a specific setting. What a machine can do under a specific setting is the intelligence of the machine programmer, not the machine itself. For example, an interactive dictionary stores a lot of human knowledge and it can do remarkable things for humans, but it is not intelligent. Test criteria for machine intelligence may also provide quantitative feedback for improving the intelligence tests for humans.

3 Predicted Impacts

The history of science and technology has shown that impressive technical improvement and persistent cost reduction will follow an important scientific revolution. The amount of technical improvement and cost reduction can be so great that it was difficult to foresee at the time of revolution. Two well known examples are the internal combustion engine technology to today's automobiles and Von Neumann machine idea and the semiconductor technology to today's popular computers. The following predictions may seem to be overly optimistic today, but the history could prove them to be true.

3.1 Human life

This revolution will greatly improve the quality of human life. The introduction of engines greatly relieved humans from hard *manual labors*. The introduction of computers greatly relieved humans from mechanical *computation labors*, especially those that humans cannot do as fast, such as doing calculations, controlling a complex machine or generating synthetic graphics images in real time. The introduction of developmental robots could relieve humans from tedious *thinking labors*. Those are low-level thinking tasks, mainly to execute human high-level commands. The quality of human life could be greatly improved with the arrival of the age of developmental robots. Developmental robots will be used as human assistants, from factories to households. Their developed "brains" are downloaded as software to be run on desk-top computers to do various tasks, from reading emails to helping children to learn. In the past, thinking robots have been only discussed in science fiction because machine thinking has not been sufficiently understood. Thinking seems a collection of internal behaviors of a *developmental being* (animal or machine) and it must be developed through autonomous mental development just like humans and higher animals. Infants think using their simple internal behaviors and adults think using their more developed internal behaviors. A robot that runs a developmental program is like a machine that writes mental program autonomously, when the developmental program interacts with the sensory information from the real world. Its developed internal behaviors represent the true thinking by a machine.

Why did all these advances not occur in the past? This is mainly because the AI field did not pay sufficient attention to, or at least was not serious about, autonomous mental development for machines until just a few years ago. Currently, all the efforts for building AI systems follow the traditional manual development paradigm, with a few recent exceptions mentioned above. With the new paradigm, human programmers are not required to write a particular program for each of the tasks that we want the machines to perform, which has been proved extremely difficult if the task requires what we consider as intelligence. Instead, what the human programmers need to do is to write a developmental program, which is of general purpose. Although developmental programs are by no means easy to design, they are easier to understand and to improve than many special systems designed for specific AI tasks. The practical aspect of developmental robots also rests in the ease of training. The user of a developmental robot does not need to write a program or manually feed data if he wants to teach the robot. He just trains the robot very much like the way he trains a human child, showing it how to do something while talking to it, encouraging or discouraging what the robot does from time to time. Thus, everybody can train a highly improved developmental robot, a child, an elderly, a teacher, a worker — anybody. This is the basic reason why the developmental robots could become popular. Computers would not have been that popular today if they are not as easy to use as today's computers with very intuitive graphical user interfaces.

3.2 Economy and jobs

The economic impact of developmental robots will be enormous. The country that takes the lead in developmental robots will first create a new industry for this new kind of machine. This new industry will take advantage of the advanced automobile industry to develop sensor-rich humanoid robots (Honda in Japan has already started it). It will also take advantage of the fast progress of the computer industry to build computers and memories best suited for the computational need of developmental robots. The cost for large storage will drop consistently when the market grows. For example, the cost of hard-disk storage that is of human brain size in terms of number of bytes has already dropped from about $5M in 1998 to around $250,000 today (June 2000). Real-time speed with large memory is reached through coarse-to-fine memory search schemes. There will be a new industry for humanoid robots, fueled by the need for building bodies for developmental robots. Many different types of bodies, designed for different working conditions and environments will be made to satisfy increased application scope of developmental robots. It is expected that in the next 10 to 20 years, the developmental robot industry will primarily aim at professional applications, such as research institutions, amusement parks, public service areas, and the defense industry. During this period, consumers can benefit from the software that is developed on professional robots. Eventually, developmental humanoid robot may cost the same as a car plus a high-end personal computer. The country that takes the lead in this new endeavor will create an abundance of economical activities and well-paid jobs related to this new industry.

3.3 Understanding of human mind

The impact on the scientific understanding of our mind will be far reaching. This revolution will drastically improve our understanding about one of the most complex subjects that faces mankind today — our own minds. For example, what are the basic mechanisms that govern the ways in which our minds develop? To what degree can the environment change the formation of the mind? What can the environment do to effectively and positively influence the human mind and improve the life of mankind? The answers to these questions require the knowledge about the developmental root of the mind. Without studying the computational models of mental development, these questions cannot be sufficiently and clearly answered.

3.4 Medicine

The knowledge created by this revolution will also improve medical care. It will provide basic knowledge useful for treating learning disabilities, mental disorders, and mental problems associated with aging. For example, what developmental mechanisms are responsible for attention deficiency? What developmental mechanisms are responsible for enabling an individual to establish the value of an event, a behavior, or the social norm? What techniques are effective for teachers to improve the development of certain cognitive and behavioral capabilities? Computationally, which areas of the brain are responsible for certain mental disorders? During aging, which mechanism of the brain is likely to deteriorate first and what remedies are possible?

4. Why now?

As we discussed above, the recent new discoveries about human brain tell us loud and clear that our human brain utilizes the developmental principles that are shared by different sensing and effector modalities. Since higher brain functions appear to be even more plastic than early sensory processing, it is expected that the higher brain functions also use developmental principles that are generally applicable to different subject matters that humans learn. The time is right to study what these developmental principles really are.

Technically, it is now possible to study massively parallel, distributed brain activities and relate them to mental development. The advances in neural imaging techniques, such as EEG, EMG and fMRI, now allow high resolution, concurrent, and real-time measurement of brain activities.

In the machine intelligence and robotics fields, the fundamental difference between the way human mind is developed and the traditional engineering paradigm for machine development was recently identified as the fundamental reason for the difficulties in AI. The studies about the fundamental limitations of the current engineering paradigm have recently started.

Some preliminary computational models for developing the mind by machines were recently proposed the tested. These early efforts have achieved some results that have not been possible using the traditional engineering paradigm. Therefore, computational models of mental development for machines are not beyond human comprehension and they are within the manageable scope for humans to model computationally.

The performance-to-cost ratio of computers has reached a critical level that now it is practical to simulate brain development in real time on a robot, with a storage whose size is equivalent to a considerable fraction of human brain. Further, this can now be done at a very moderate cost. For example, the development of the most computational challenging modality, vision, can now be simulated on real robot in real time by software running on a PC workstation.

Technology for building robots has also been improved significantly. In recent years, research laboratories and related industries in US and Japan gained remarkable experience in actually building robots that resemble human and animal bodies with similar articulate structures, from human-size humanoid robots (e.g., the series of Honda humanoid robots) to advanced consumer toy robots (e.g., Sony AIBO dog robots). The robotic technology is ready for building various humanoid or animal robots as bodies for developmental machines.

5 Research issues

In some sense, the task-nonspecific nature of mental development makes the studies of mental development easier than the traditional task-specific approaches. This is true for both human subject (neuroscience and psychology) and machine subject (AI and robotics). From the computational view of mental development, the research issues are around sensory signals and effector signals with internal autonomously generated numerical states. A developmental program will associate signals that are from different sensors, stored in internal status and sent to effectors, but its programmer does not need to know what those signals actually mean! To put it intuitively, it is easier to model how an interactive program looks up words from its word memory than to model how the meanings of words in The Merriam-Webster's Dictionary relate to one another. The former is like what a developmental program does for many tasks that a developmental being will come across and the later is like what all the traditional programs do for a particular task.

To understand this fact better, we take a complex behavior as an example. Modeling attention selection in a traditional task-specific way requires the researcher to understand the nature of the task (e.g., driving) and then to study the rules of attention selection based on the steps of the task. Such rules are extremely complex (e.g., due to the complex road situation during driving) and the results are ad hoc in the sense that they are not directly applicable to other tasks or even to the same task under different scenarios. In contrast, attention selection by a developmental being is just a part of behaviors that are being developed continuously and constantly. As long as the effectors for attention selection are defined for the body (external effector) and the brain (internal effector), the attention selection principles are developed autonomously by the same developmental program in a way very similar to the behaviors for other effectors, such as arms and legs.

Consequently, a series very interesting and yet manageable new research problems are opened up for study, for fields that have either human or machine as study subjects. Some of the tractable research problems that can be immediately studied are suggested below.

1. Schemes for autonomous derivation of representation from sensory signals (from the environment and the body).
2. Schemes for autonomous derivation of representation from effector signals (from the practice experience)
3. Autonomous derivation of receptive fields, in both the classic and nonclassic sense. That is, how later processing elements in the brain group outputs from earlier processing elements or sensory elements.
4. Long term memory growth, self-organization and retrieval, for high-dimensional neural signal vectors.
5. Working memory formation and self-organization, for high-dimensional neural signal vectors. The working memory may include short term sensory memory and the system states.
6. Developmental mechanisms for mediation of conscious and unconscious behaviors. That is, those for mediation among higher

and lower level behaviors, such as learned behaviors, learned emotional behaviors, innate emotional behaviors and reflexes.

7. Mechanisms for developing internal behaviors — those that operate on internal nervous components, including attention selection. This subject includes both developmental mechanisms and training strategies for humans and robots.
8. Attention-directed time warping from continuous states. This subject deals with time inconsistency between different instances of experience, with the goal of both generalization and discrimination.
9. Autonomous action imitation and self-improving. The developmental mechanisms for a developmental being to derive an improved behavior pattern from individual online instances of related experience.
10. Mechanisms for *communicative learning* and thinking. The developmental mechanisms that allow later learning directly through languages (auditory, visual, tactile, written etc) as children do when they attend classes. These mechanisms enable development of thinking behavior, which is responsible for planning, decision making and problem solving.

6 Performance metrics

The current fragmentation landscape of AI is a reflection on how different AI problems can be measured by very different metrics, if intelligence is measured as the capability of performing a specific task. However, what a machine can do under a specific setting represents the intelligence of the machine programmer, not necessarily the machine's own intelligence. Further, a special purpose machine that can only work for a particular problem cannot deal with complex problems that require true intelligence, such as vision, speech and language capabilities.

The *criteria for measuring machine intelligence* will fundamentally change. The metrics that can be used to measure the power of developmental robots should emphasize the autonomous interactive learning capabilities in complex human environment. In other words, it is the capability of mental development instead of what the machine can do under a pre-specified setting.

This is indeed the case with well-accepted test scales used by clinical psychologists for measuring mental and motor scales of human children. Two such well known scales are The Bayley Scales of Infant Development (for 1 to 42 months old) and The Leither International Performance Scale (for 2 years to 12 years old). These scales have a very systematic methodology for the administration of tests and scoring. The reliability and calibration of these scales have been supported by a series validity studies, including constuct validity, predictive validity, and discriminant validity that cover very large number of test subjects and different age groups across very wide geographic, social, and ethnic populations.

Here let us take a look at an example of tests in the Leither International Performance Scale for a two years old. The name of the test is Matching Color. The test setup is a row of 5 stalls. Above each stall pasted a color card, black, red, yellow, blue, and green, respectively. During the test, color blocks are presented, one at a time in the order: black, red, green, blue, and yellow. The examiner places the black block in the first stall and tries to get the subject to put the red block in place by placing it on the table before him, then in the appropriate stall, then on the table again, nodding to him to do it and at the same time pointing to the second or red stall. As soon as the subject begins to take hold of the test, the final trial can be attempted. In this test, the examiner tries to get the subject to imitate his procedure. The test is scored as passed if the subject is able to place the four colors (the first one is placed by the examiner) in their respective stalls entirely by himself during any one trial, regardless of the number of demonstrations or the amount of help previously given by the examiner. As we can see, the test does not really concern about whether the child has learned the abstract concept of color, but rather the capability of imitating the action of the examiner using visual color information as a cue in coordination of his motor effectors (hand and arm).

The mental age that is used for measuring human intelligence in these tests can be used as a scale for measuring machine intelligence. Currently metrics that have been used for various AI studies mainly measure what a machine can do under a specific setting, instead of the capability of mental development. Such a capability requires online, interactive learning capability as the above test demonstrates. For example, an interactive dictionary stores a lot of human knowledge and it can do remarkable things for humans, but it is not intelligent. If a machine that can pass the systematic tests like the one shown above, it must have already learned many others skills that no traditional machine has. Therefore, although autonomous mental development is a new direction, its impact on the future of machine intelligence and our understanding of human intelligence will be far reaching. The performance metrics for measuring intelligent machines can be adapted from those used by clinical psychologists for testing the mental development of human infants. The Bayley Scales of Infant Development and The Leither International Performance Scale are two such examples.

References

[1] T. S. Kuhn. *The Structure of Scientific Revolution*, University of Chicago Press, third addition, page 68, 1996.

[2] L. von Melchner, S. L. Pallas and M. Sur. Visual behavior mediated by retinal projections directed to the auditory pathway. *Nature*, vol. 404, April 20, pages 871-876, 2000.

[3] X. Wang, M. M. Merzenich and K. Sameshima and W. M. Jenkins. Remodeling of hand representation in adult cortex determined by timing of tactile stimulation. *Nature*, vol. 378, no. 2, pages 13-14, 1995.

[4] J.L. Elman, E. A. Bates, M. H. Johnson, A. Karmiloff-Smith, D. Parisi and K. Plunkett. *Rethinking Innateness: A connectionist perspective on development.* MIT Press, Cambridge, MA, 1997.

[5] E. Thelen, E., G. Schoner, C. Scheier, and L.B. Smith (In press). The dynamics of embodiment: A field theory of infant perseverative reaching. *Behavioral and Brain Sciences*, to appear.

[6] J. Weng. The Living Machine Initiative. Department of Computer Science Technical Report CPS 96-60, Michigan State University, East Lansing, MI, Dec. 1996. A revised version appeared as a chapter: J. Weng. Learning in Image Analysis and Beyond: Towards Automation of Learning, in *Visual Communication and Image Processing*, C. W. Chen and Y. Q. Zhang (eds.), Marcel Dekker, New York, NY, 1998.

[7] N. Almassy, G. M. Edelman and O. Sprons, Behavioral constraints in the development of neural properties: A cortical model embedded in a real-world device. *Cerebral Cortex*, vol. 8, no. 4, pages 346-361, 1988.

[8] J. Weng, W. S. Hwang, Y. Zhang and C. Evans, Developmental robots: Theory, Method and Experimental Results, in Proc. 2nd Int'l Symposium on Humanoid Robots, Tokyo, Japan, pp. 57- 64, Oct. 8- 9, 1999.

[9] D. Roy, B. Schiele, and A. Pentland. Learning Audio-Visual Associations using Mutual Information. In Proc. *International Conference on Computer Vision, Workshop on Integrating Speech and Image Understanding.* Corfu, Greece, 1999.

[10] M. Huber and R. A. Grupen. A feedback control structure for on-line learning tasks. *Robotics and Autonomous Systems*, vol. 22, no. 3-4, pages 303-315, 1997.

[11] H. Gardner. *Multiple intelligences: The theory in practice.* Basic Books, New York, NY, 1993.

PART II
RESEARCH PAPERS

6. LEARNING: TOWARD SELF-EVOLVING ARCHITECTURES

Evolution of Intelligent Systems Architectures: What Should Be Measured?

A. Meystel
Drexel University, Philadelphia, PA 19104

Abstract

Various degrees of intelligence evolve in the intelligent systems as a result of their development by the virtue of external design and/or self-organization. The increase in degree of intelligence is achieved via evolution of its architecture. This paper is intended to establish a conceptual and methodological background required for design and evaluation of performance and the degree of intelligence of intelligent systems. The paradoxical ability to increase redundancy while reducing complexity is described as a hallmark of intelligence. The naturally evolved architectures of intelligence are constructed in such a manner that the tools of complexity reduction do not curb the combinatorial capabilities of the system.

1. Intuitive Approaches to the Concept of Intelligence

An attempt is made to approach the concept of intelligence constructively and from the scratch. This analysis is motivated by the need for using the results for *constructed* (primarily. cngineering) *intelligent systems* and *agents*. In the author's view. the Descartes` problem (of the Mind existing separately from the Body) simply doesn't exist. because the Mind of the constructed Machine is undoubtedly produced by its physical components ("body"). Yet all phenomena of intelligence in living creatures seem to allow for their computational modeling. Nevertheless. the author doesn't adhere to the technological paradigm alone. Both the examples of intelligence and its architectures will be discussed for all domains shown in Figure 1.

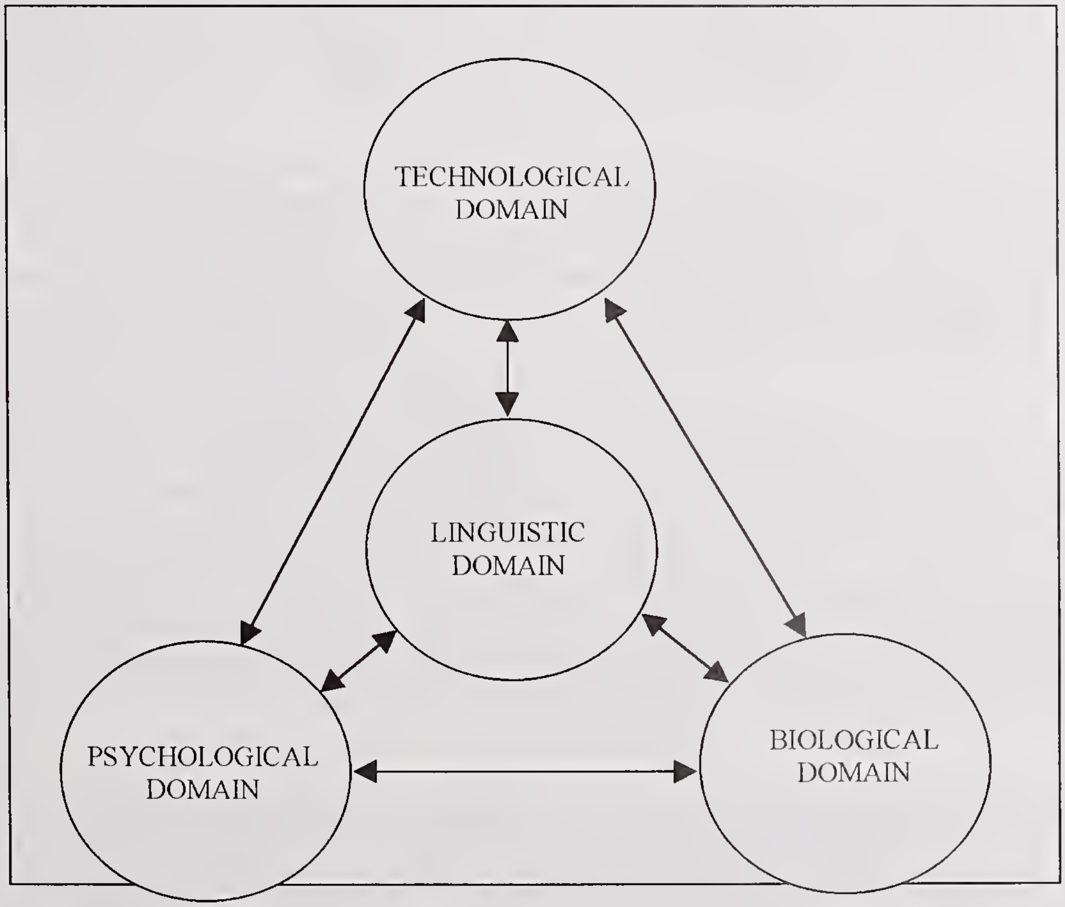


Figure 1. Techniques linked with and stemming from the concept of *intelligence*.

The goal to construct the architectures of intelligence and analyze their evolution can be achieved if a comprehensive definition of intelligence is introduced. It seems meaningful to derive the definition from integrating the phenomena characteristic for intelligence. Obviously, they can be demonstrated in relevant systems belonging to all domains shown in Figure 1. Interestingly enough, within each of these domains, there are common habits of discussing intelligence. Possibly, this is a result of the fact that all of them depend on the linguistic domain. The main habits of talking about intelligence can be listed as follows:

1. Functioning of intelligence is frequently characterized in the anthropomorphic terms of mental conduct.
2. Intelligent activities are attributed to levels of generality (levels of *scope*)

1.1 Features of Mental Conduct

The terms of natural language that characterize intelligence both positively and negatively, can be used for evaluating the richness of the concrete domain of discussion and judging whether domains from Figure 1 are well represented. One can make an observation that all of these properties can be quantitatively evaluated in a crisp or fuzzy manner.

Table 1. Antonyms characterizing Intelligence (From [1])

clever	⇔	dull	observant	⇔	unobservant
sensible	⇔	silly	critical	⇔	uncritical
careful	⇔	careless	experimental	⇔	unexperimental
methodical	⇔	unmethodical	quick-witted	⇔	slow
inventive	⇔	uninventive	cunning	⇔	simple
prudent	⇔	rush	wise	⇔	unwise
acute	⇔	dense, obtuse	judicious	⇔	injudicious
logical	⇔	illogical	scrupulous	⇔	unscrupulous
witty	⇔	humorless	smart	⇔	stupid

The tendency to using these adjectives for characterizing intelligent systems in all domains is unavoidable. Although, they could be called anthropomorphic, their use seems to be justified even as applied to living creatures different from humans such as apes, cats, dogs, horses, mice. Then, we might agree with using at least some of these terms to analyze intelligence of birds, fishes, reptiles. After getting used to see the common patterns we can expand some terms related to intelligence into domain of insects, and then, proceed toward bacteria. too.

Analysis of the intelligence related vocabulary helps to discover a number of other phenomena that should be taken in account in constructing definitions and models for *intelligence*. Indeed, from the fact that **stupidity ≠ ignorance** we can conclude that **intelligent ≠ possessing knowledge**. Thus, **having knowledge**, or **being informed** could not be considered a base for defining *intelligence*. On the other hand, *intelligence* is frequently associated with a comparably vague concept of the activity of **thinking**. The latter contains as a part, such activity as **theorizing**, and one can expect that **theory formation** should be represented in the architecture of intelligence. It is the capacity for creation of a rigorous theory that lays the superiority of men over animals not the capacity to attain knowledge.

It would be desirable to embark on constructing the definition of intelligence focusing upon most of the factors that is linked with this complex phenomenon of mental conduct. Before introducing architectures of intelligence a set of mental conduct epithets was analyzed including such terms as:

careful stupid logical unobservant ingenious
vain methodical credulous witty self-controlled
and their correlations so that they could be represented in the definitions and architectures..

1.2 *Intelligence* is a Property existing at all levels

In addition to multiple properties and phenomena related to the mental conduct, intelligence invokes talking about *level of intelligence*. The term intelligence is attributed to each of the interrelated levels including

$$\left.\begin{array}{l}\bullet\ \text{societal phenomena}\\ \bullet\ \text{group activities}\\ \bullet\ \text{individual activities}\\ \bullet\ \text{organ functioning}\\ \bullet\ \text{cell functioning}\\ \bullet\ \text{DNA functioning}\end{array}\right\}\quad \text{scaled by the unit of } \textit{intelligent agent} \qquad (1)$$

These levels are apparently associated with the scale (resolution, granularity) of representing the external reality by the functioning intelligence. Some of these levels emerged because humans introduced them. Some of them evolved naturally (biologically, ecologically, or psychologically). In all cases, the multiresolutional organization improves the efficiency of functioning [2]. Each particular level of resolution is scaled by the nature of the hierarchy (1). At the same time, for each agent within the hierarchy (1) another multiresolutional scaling can be introduced for units of interest existing within a level and requiring its own hierarchy of levels that makes operations with this unit more efficient. It seems that the ability to come up with a multiplicity of levels of resolution is a property of intelligence that produces these levels of resolution.

On the other hand, each of the levels mentioned above can be characterized by the ability to build and construct rules associating objects and activities at the level, and by the ability to introduce and use theories. Both rules and theories are formulated by the researcher observing and analyzing external intelligence. However, they reflect the properties and laws existing within the system of objects under consideration. Both rules and theories are applicable for the decision making processes that are utilized to control

- objects at a level
- levels as a whole
- the overall system that is combined out of levels and contains these objects.

Let us notice that the organization of the system to be controlled is affected by the intelligence, and the introduction of rules and theories is done by the intelligence, too. The source of the intelligent in both cases is not determined, and the intelligence as a phenomenon is undefined.

2. Introducing Formal Approaches

2.1 General Statements

It looks like the Theory of Control that does not take in account the phenomenon of intelligence, is not fully equipped for solving problems for the domain of intelligent systems, e.g. in robotics. The particular problem is in determining VECTOR OF INTELLIGENCE OF A CONTROLLER and putting it in a correspondence with the VECTOR OF PERFORMANCE. Designers are dealing with systems that are underspecified even as far as their *inputs* and *outputs* are concerned.

Thus, the first two emerging questions are: 1. What are the inputs into the system under consideration? and 2. What are the outputs of this system? The input can be introduced by the designer of the architecture and by the values of variables provided by a specific architecture (intelligence). The output is always understood in the terms of performance.

An attempt to answer the questions requires to revisiting the logical categories that are used for analysis of systems with intelligence (in particular, *intelligent control systems*). Simultaneously, we must determine whether we will discuss these issues in the terms of predicate calculus of the first order, in the terms of other logical systems, or in the terms of meaning extraction and interpretation of the Natural Language. The lists of inputs and outputs contain concepts that entail a diversity of various schemes of reasoning and logical categories. The logical type of category, to which a concept belongs is a set of ways, in which it is consistent, i.e. it is logically legitimate to operate with. To determine a logical network of concepts is to review the logic of propositions, in which they are utilized including the following:

1. With what propositions of the classical control theory, the propositions related to intelligence are consistent and/or inconsistent
2. What are the new propositions of control theory that follow from the propositions related to intelligence

One of the challenges is to determine whether for the alternative definitions of intelligence and the associated processes we selected correctly the logical categories in terms that are consistent with the practice of design and application in the domains shown in Figure 1. In particular, we would be interested whether the concepts of thinking, mental powers, smartness, their components and operations they entail have been coordinated consistently. It should be demonstrated that there is no operations with these concepts and processes that breach logical rules. We suspect that the consistent system can be built if the logical consistency will be determined not in the terms of predicate calculus of the first order but in the terms of laws of interpretation determined for the Natural Language used for describing the real systems and situations.

2.2 List of Premises that Are Characteristic for Intelligent Control

The following premises can be considered as following from the experiences in all domains of Figure 1 in the cases of exploring intelligent systems as objects and intelligence as a phenomenon of these objects.

2.2.1 Cultivating redundancy is a prerequisite of intelligence

Redundancy of systems is understood as having their resources, components, or properties in abundance, or in excess. It is a feature of intelligent systems that information they deal with is intrinsically redundant and the tools of processing this information are in excess of the minimally required set of tools. It is a feature of intelligent systems to cultivate this redundancy and it will be shown that intelligence is equipped by specific tools for doing this.

This property of redundancy is very important and very characteristic for intelligent systems. They should be always ready to withstand uncertainty, and since the survival is at stake, the property of redundancy helps to minimize the risk of failure. E. Ruspini has mentioned: the systems should have more intelligence than it needs for solving the problem[1]. Obviously, the same problem can be resolved with different level of intelligence. Then, the results of this problem-solving process could be used for evaluating the level of intelligence. This level might depend on the level of redundancy.

Although redundancy as a property is considered negative (it should waste resources), ot only intelligent systems do not fight redundancy, it explore, use, and even cultivate the redundancy. Redundancy is the tool for combining and testing new alternatives of decisions. After evolving intelligent systems develop a mechanism of exploring things within its "virtual reality," redundancy is becoming a tool for planning and a tool for learning without actually having physical experiences.

Autonomous systems should acquire info in physical (realistic) and/or imaginary playgrounds. The following factors are being displayed related to redundancy:

[1] In an exchange during the panel on Intelligent Control at IJCNN'2000, E. Ruspini commented that probably such creature as E.coli possesses all intelligence it needs for functioning. A. Meystel proposed a paradoxical circular definition for intelligence that illustrates and further develops Ruspini's statement: "The system is intelligent iff it has more intelligence than it needs."

- — Playfulness is a property observed in living creatures or linguistic systems that are characterized by a very high level of intelligence. Playfulness of an intelligent system is to be considered a part of the learning process.
- — Redundancy supports various manifestations of the property called "desire" including all known classical desires that determine foraging and reproductive activities.
- — Certainly, speaking about "playful ameba" might be a stretch however searching activities are observed even for amoebas [5] and E.coli's [6] (and this allows to talk about certain degree of intelligence even in these classes of living creatures [7].

Intelligent systems are equipped by multiple tools of acquiring and increasing their redundancy. Learning is one of the tools that employs actual experiences or imagination.

2.2.2 Reduction of complexity is a working technique of intelligence

How is it possible to cultivate redundancy, and yet fight complexity? This paradoxical ability is a hallmark of intelligence. Practically, it means that the tools of complexity reduction should not curb the combinatorial capabilities of the system. Such tool exists, and this is organization of information in a multiresolutional fashion (see [3, 4]). This organization of information actually determines appearance of the levels mentioned in sub-section 1.2.

The need to evaluate and reduce complexity was always clear in computational mathematics and this led to the concept of epsilon-entropy and techniques of its evaluation [8]. Many elegant mathematical techniques of complexity reduction has been developed (e.g. like in [9]). The specifics of application domain was appreciated (see [10] for the software complexity, [11] for syntactic complexity, [12] for complexity of information extraction, [13] for information of control system).

However, the need to use multiresolutional organization of information for complexity reduction was not immediately acknowledged and considered an understandable and desirable tool even after publication of [3, 4]. Further explanation of relations between multiresolutional tools of complexity reduction can be found in [14, 15].

In all systems (technological, biological, psychological and linguistic) formation of multiresolutional representation is a technique of complexity reduction. Even E.coli fights the complexity by forming at least two levels of resolution (high resolution – single E.coli, low resolution – swarms formed as a result of bacteria gathering in groups [7]).

2.2.3 Loop of Semiotic Closure is the Primary Architecture of Intelligence

The modules of (1) World, (2) Sensors, (3) Perception, (4) World Model, (5) Behavior Generation and (6) Actuators, connected in a loop of closure, are forming an Elementary Functioning Loop, or ELF. The module of World is the ambient environment including a source of information from the process generated by Actuation to be observed by Sensors. This component of the World also consumes the energy submitted by the module of Actuation. If one interprets Figure 2 as a general structure of an intelligent vehicle, then the module of World is the couple Vehicle/Road. The energy is conveyed through this couple to the body of the moving Vehicle, the vector of speed is measured for the Vehicle relative to the Road within this couple. Sensors are transducing the information from the domain of physical reality to the information carrier accepted by the system of computation. In addition, Sensors are responsible for complicated activities linked with organization and coordination of testing. These activities are a part of another loop of closure (see [17]).

The module of Sensory Processing organizes the information and submits it to the World Model that puts the units of acquired knowledge into a form appropriate for storing and utilization by the module of Behavior Generation. The latter may vary from the simple look-up table to the complex devices that explore alternatives of plan and simulate them before submitting them to the module of Actuation. The simple look-up table would contain the list of control functions f(t) together with previously experienced or expected measures of achievement J (f, x, x*) for the given goals x*(t) and present situations x(t) as couples

$$x^*(t), x(t) \rightarrow f(t), J(f, x, x^*). \qquad (2)$$

The concept of semiotic closure is not an obvious one. It exceeds the straightforward idea of feedback that can be formulated as follows. In a system, there exists a monitor (human or electronic/mechanical) that compares what is happening at time t, x(t), with some standard of what should be happening x*(t). The difference or error, Δ(t) = x(t) – x*(t), is fed to a controller for generating an action by a control function f(t)=y(t+k), which can be taken only at a later time, t + k. Thus, the feedback equation presumes some

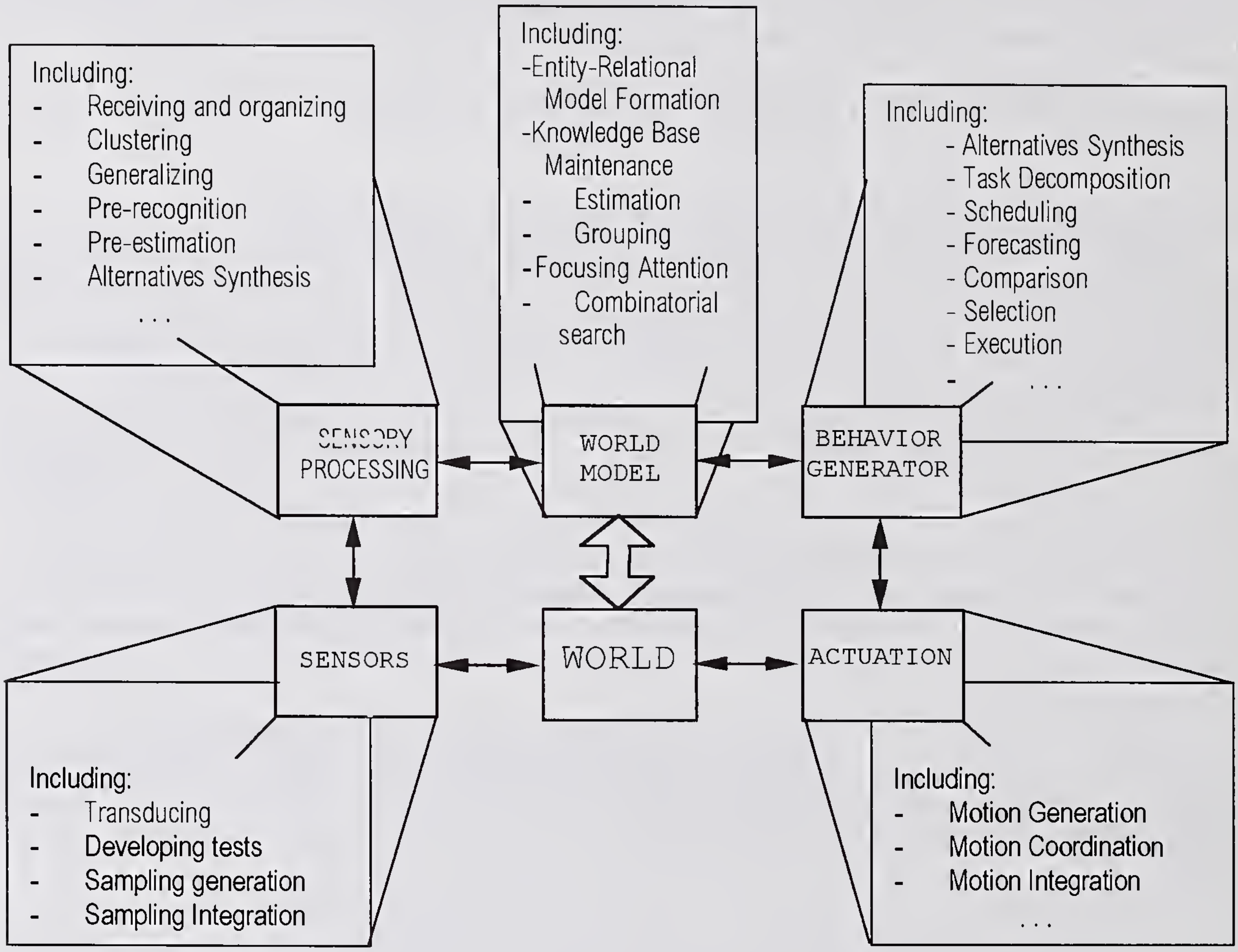


Figure 2. Semiotic Closure for a System With Motion

$$y(t + k) = f\,[\Delta(t)] = f\,[x(t) - x^*(t)] \qquad (3)$$

standard assigned, some variable compared with this standard, and some device that computes "feedback compensation." The standard might be assigned as a goal externally, or stored in the module of World Model. The device that computes "feedback compensation" can be associated with the module of Behavior Generation. Sensors, Sensory Processing and World are meant but not explicated. Certainly, this concept should be enhanced substantially to be transformed into the concept of semiotic closure.

Semiotic closure was anticipated in 1967 by L. von Bertalanfy [18] who considers feedback to be "a special case of general systems characterized by the presence of constraints which led the process interpretation toward **circular causality** and thus making it self-regulating. This loop of "circular causality" was dubbed "semiotic closure" by H. Pattee in 1973 [19]. It was introduced to analysis of intelligent systems in [20] and [21]. Semiotic closure can be constructed for any domain and any system that exhibit elements of intelligence.

2.2.4 Entity-relational network (ERN) is a frequent form of constructing the representation at a level of intelligent system

It would be more prudent to say that we simply do not know any alternative to ERN. Of course, we can approximate ERN by a multiplicity of tables and approximate each of the tables by an analytical function. We do this for the variety of manual activities. However, as computer permeates our workplace, we found that having ERNs even in a tabular form is the most flexible way of storing information.

Thus, a problem of generalization emerges as a problem of local substitution of large accurate tables by small tables with larger but still acceptable error. Thus, instead of a global gigantic ultimately accurate ERN, we receive a set of entity-relational networks $\{ERN_i\}$, i=1,2,..., n where 1 is the index (number) of the level with highest resolution, n is the index of the level with the lowest resolution. The system does not have all these levels in its storage because the amount of information in $\{ERN_i\}$ would substantially exceed the amount of information in its level of highest resolution ERN_1. The system remembers only levels with middle (average) resolution and selected traces at the level of higher and/or lower resolution. If it requires more lower resolution information, it generalizes the middle level information as necessary. If it requires higher resolution information, it instantiated (decomposes) the information top down as requested. The system $\{ERN_i\}$ is a nested system, i. e. the conditions of inclusion should be satisfied for the ontologies constructed for the Worlds represented at each particular level of resolution. The same conditions should be realistically satisfied for the objects and actions represented at the levels. Such a system can exist if it is supported by the operators of grouping, focusing attention (selection), searching for combinations of interest (combinatorial search), and the operators that ungroup, defocus and eliminate the results of search.

2.2.5 Constructing Multiresolutional Representation is a tool of intelligence

Each level of representation has granularity that is a result of generalizing information from the lower level of higher resolution [16]. Both objects and actions of the real world have their representatives at several (at least at two) levels of resolution and therefore are multiresolutional. The mechanism of obtaining lower resolution objects and relationships out of higher resolution objects and relationships is called generalization.

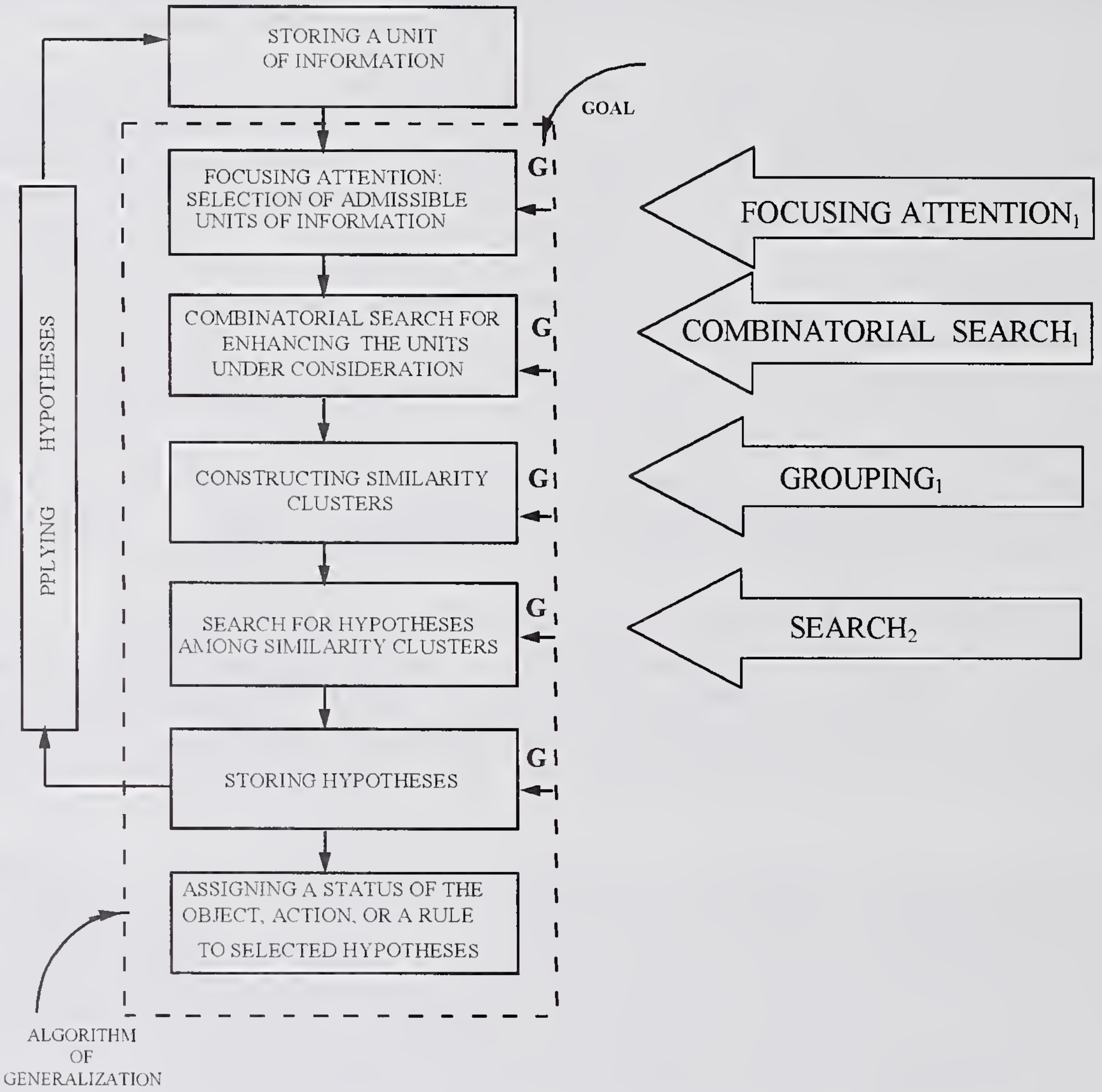
The nature of generalization was envisioned by gestalt psychologists [22]. The need in the computational theory of generalization was emphasized by J. McCarthy in [23]. One of the possible algorithms of generalization is demonstrated in Figure 3. One can see in this example that the algorithm consists of operators that perform Grouping (G), Focusing Attention (FA) and Combinatorial Search (CS) together (the subscript means the level it works for). The joint set of operators G, FA, and CS we will call GFACS. Using this set: computational procedures of grouping focusing attention and combinatorial search (GFACS) is inevitable in an intelligent systems because the level of generalized information cannot be built otherwise. GFACS generalizes information bottom up. Decomposition top down requires for an algorithm of instantiation ($GFACS^{-1}$). There exist a vast multiplicity of algorithms belonging to the class of GFACS: e.g. ARMA (auto-regressive moving average) as in [24, 25]; CMRA (convex multiresolutional analysis) as in [26] and other. CFACS-1 has its prototypes, too, such as Sieve Decomposition algorithm [27].

Encoding of stored information is done in a multiresolutional fashion too, and this leads to the further reduction of complexity because instead of storing the body of the message (the file) we can store onle the code and apply to this code the mechanism of restoring the body. It is a legitimate mechanism of storing informational entities by storing the code and regenerating (reconstructing) the information as necessary. Storing information in the form of DNA is an example of reconstructing the multiresolutional system of a living organism.

2.2.6 Cost-functional

The need in a reduction of computational complexity would be easy to resolve by abandoning computation. Yet, this cannot be done because the system has a goal to fight for reducing the time and energy that are required to reach the target. This determines the conditions of the optimization process. The latter should be performed in correspondence with the calculus of variations and Euler-Lagrange equation. The central problem that emerges is to determine properly the Hamiltonian of the system, or its cost-functional.

As far as computational complexity is concerned, the results of optimization are driving the process of forming levels of resolution. The optimization for an E.coli sounds like working under the heuristically introduced cost-functional of foraging (see [6]):



STORING A UNIT OF INFORMATION
GOAL
FOCUSING ATTENTION: SELECTION OF ADMISSIBLE UNITS OF INFORMATION
G
COMBINATORIAL SEARCH FOR ENHANCING THE UNITS UNDER CONSIDERATION
G
CONSTRUCTING SIMILARITY CLUSTERS
G
SEARCH FOR HYPOTHESES AMONG SIMILARITY CLUSTERS
G
STORING HYPOTHESES
G
ASSIGNING A STATUS OF THE OBJECT, ACTION, OR A RULE TO SELECTED HYPOTHESES
PPLYING HYPOTHESES
ALGORITHM OF GENERALIZATION
FOCUSING ATTENTION₁
COMBINATORIAL SEARCH₁
GROUPING₁
SEARCH₂

Figure 3. An Algorithm of Generalization and the Essence of its Operations

$$J = \frac{E_{consumed} \pm E_{lost}}{t_{curr} - t_0} \tag{4}$$

or

$$J = \frac{E_{consumed}}{E_{lost} \cdot (t_{curr} - t_{.0})} \tag{5}$$

Using (4) and (5) for performance evaluation is a not a very simple matter. The system might actually have many cost functionals pertaining to different levels of resolution. This can entail mutually conflicting processes of optimization. Therefore searching for an optimum motion trajectory in the multiresolutional state space would require recursive top-down/bottom-up algorithm of searching.

2.2.7 Ability to recognize and achieve goals

This ability should be considered an absolutely distinct feature of intelligent systems. In the simple artificial intelligent systems, only the highest goal (belonging to the lowest level of resolution) should be assigned to the ELF. The other goals will be obtained autonomously as a result of the planning process. Searching for an optimum motion trajectory at each level of resolution should be performed under a particular goal assigned for this level of resolution. In the case of intelligence for mobile autonomous vehicles a concept of *horizon of goal assignment*, or *horizon of planning* seems to eliminate many difficulties in developing multiresolutional algorithms of behavior generation. The concept of "horizon" is introduced because of the following conjecture:

Conjecture of Reduced Accuracy for Remote Objects and Events

> Under the same conditions and assumptions about the units of knowledge stored in the system of representation, the units that are remote spatially or temporally from the current state should be assigned lower accuracy because the risk increases of being affected by the sources of uncertainty.

As a result, the higher the resolution is the smaller is the horizon of goal assignment. Thus, from the results of finding the optimum trajectory of motion at low resolution, an intermediate state of this trajectory should be chosen as the intermediate goal-state for the level of higher resolution.

2.2.8 Emergence of "Self"

Discussions about intelligence are permeated by the statements related to "consciousness." This paper decouples the issue of intelligence and the issue of consciousness by introducing the concept of representing "self." The need in representing self arises at some level of early learning processes because of the need to increase the efficiency of planning [28]. At the initial stages of development of the robot intelligence, the whole World Model is being constructed relative to the robot. It is always situated in the center of the state representation. As the knowledge gets more complicated, the need emerges in representing the system in coordinates associated with the external system. This leads to a discovery similar to that known as the Copernicus Revolution (apparently, Ptolemy failed to put the "self" on the map, and this made using of his system less efficient).

The "self" emerges for an intelligent system (IC) after the representation of the IC itself becomes a part of the World Model constructed by IC and thus, the model of IC is shown within its own system of representation. Thus, the whole model of system (ELF, earlier shown in Figure 2) should emerge within the World Model as shown in Figure 4.

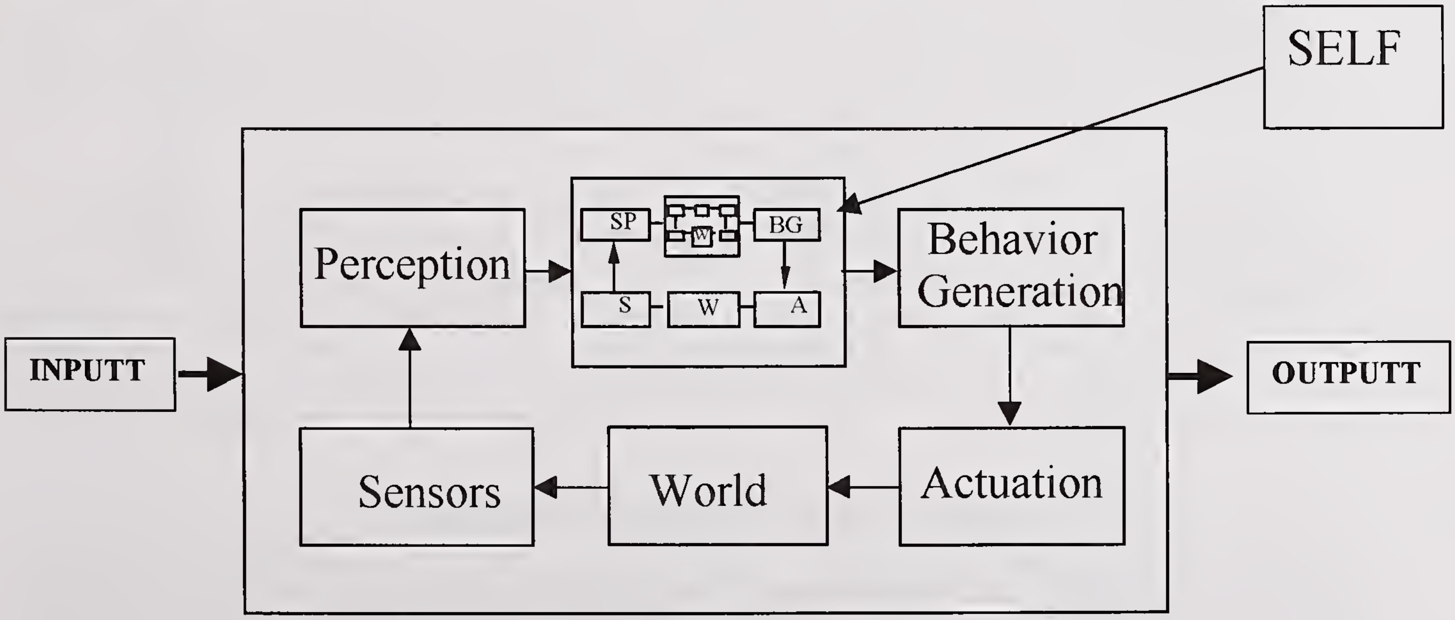


Figure 4. ELF with "self"

If IC constructs its own ELF within its representation, it should have within its World Model a representation of itself, too. Thus, the idea of "self" leads immediately to a paradoxical demand of having within its system of representation an infinite system of nested models. Obviously, this is practically impossible. Of course, in practice one or two nesting would be totally sufficient.

However, this is not the only paradoxical effect that can be listed for this phenomenon (called "reflexia" in scientific psychology). The situation gets more complicated when the World Model should also include the model of another intelligent system (IC) of a comparative level of intelligence. Then, the representation of another IC should include its representation of the first IC, which contains in its representation the first IC with its representation of both the first and the second ICs.

One of the important consequences of the emergence of "self" is that a communication with this intelligent system is possible as if it would be an external system. Since the differences in World Models are possible between the initial ELF and the ELF of "self," this inner self might have subtle differences in the decision making process. Algorithms of communications with "self" seem to be an interesting part of introspection, particularly, of "imagination."

2.2.9 Imagination

This "self" can be considered a part of some more mundane processes that are known for many animals: the processes of imagination. Creation of "virtual reality" within our brains and supporting the decision making process by exploring alternative mentally seems to be a very powerful mechanism of intelligence increasing the efficiency of functioning. In artificial intelligent systems, "imagination" is synonymous with simulation of anticipated situations during the decision making.

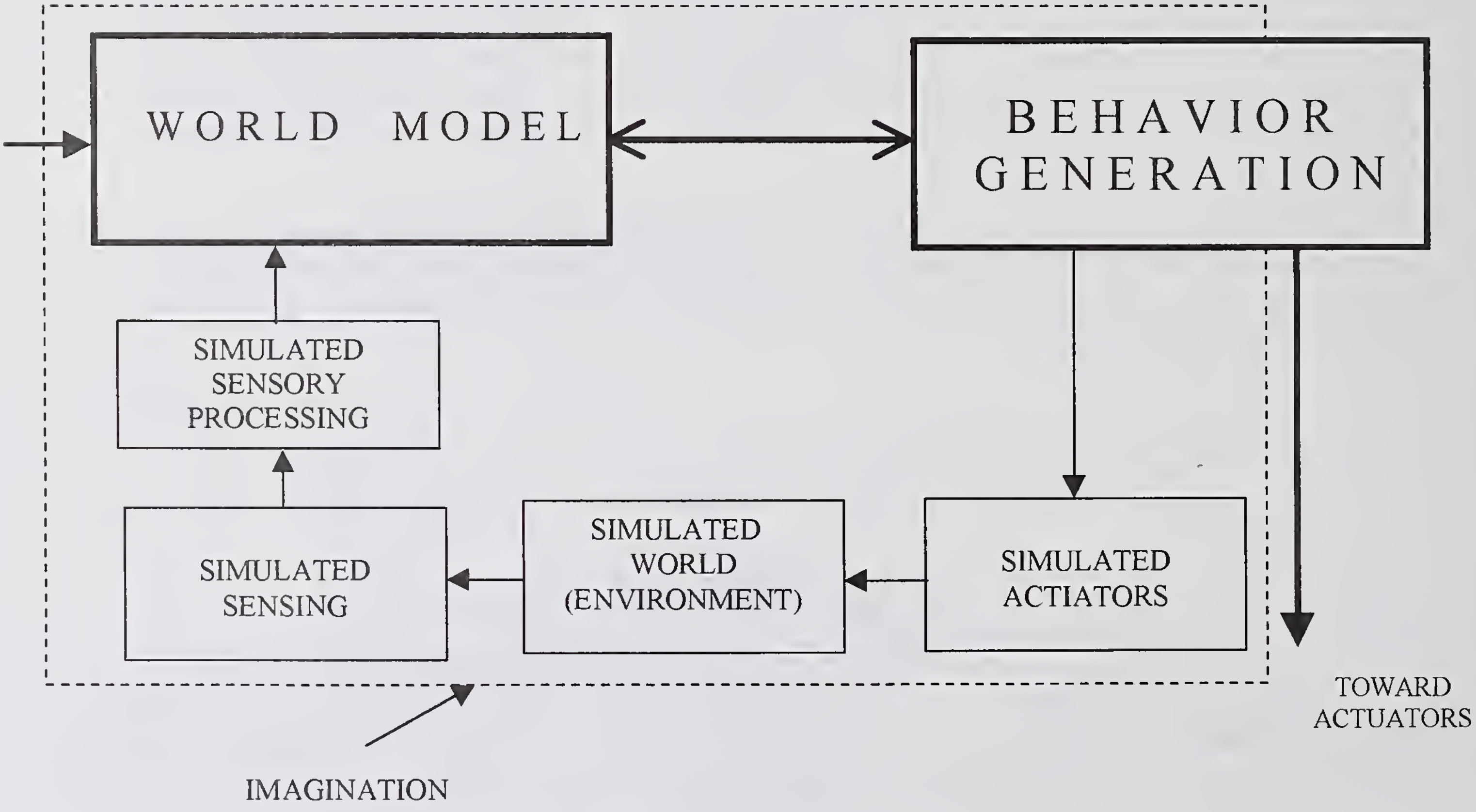


Figure 5. The system of Imagination emerging in the ELF

As Figure 5 demonstrates, instead of submitting the decision to the real actuators, the module of Behavior Generation submits it to the model of actuators, and simulates all consequences of this "WHAT IF" contemplation. IC simulates the events in the World, their development and simulates what will sensors deliver, and how sensory processing will work, and what will happen after new information is delivered to the World Model. Searching with simulating the consequences is a powerful tool of the intelligence, it is utilized for learning, planning, etc.

2.2.10 Autonomy

This property is frequently considered a synonym of "intelligence" since both of them presume each other. However, an objection is raised often that very autonomous systems can have low intelligence while very intelligent systems can be deprived of autonomy. Further analysis shows that the latter statement is not correct. If the system has low intelligence its autonomy is very limited within the world containing many systems of high intelligence. On the other hand, if the system has a high intelligence, it would require a multiresolutional level of the effort to deprive it of autonomy. This means that it would require having other highly intelligent systems to curtail the autonomy of another highly intelligent IC. Frequently, introducing the autonomy constraints happens only for the one particular level of resolution.

The important implications for multiagent systems can be expected if this topic is pursued scientifically. At the present time there are many groups that pursue the research on autonomy of multiple agents. However, not too much research is conducted about multiple multiresolutional agents. One of the important issue is the following: how much should all agents-levels worry about cost-functions of each other taken in account that they are nested within some of them while other agents-levels are nested within them.

3. Terminological Notes

3.1 Complexity

The term *complexity* is used in this paper in the following meaning: complexity is the property of a situation to consist of excessively large number of a) objects, b) relations between the objects and c) registered and modeled processes that include these objects and relationships as components, and d) unmodeled processes that depend on the stochastic factors and cannot be reliably modeled. The number should be considered excessively large if as a result of its value the cost-function that evaluates the goodness of the activities deteriorates. Evaluation of the number of components or connections, or processes, or all of the above factors should reflect the following facets of the performance:

- time of computation,
- reliability of functioning, or
- probability of emergence of the phenomena unaccounted for in the logical analysis.

In many recent publications there is a tendency to associate the term complexity only with the latter phenomenon from the list above (unmodeled processes). These references to something generated by complexity but difficult to model are actually references to the lack of knowledge of what is going on. Thus, in this paper we refer only to the phenomena "a" through "c" from the definition above.

3.2 Reasoning

The term *reasoning* is understood as applying all or most of the rules consistently and directed toward the goal. Consistency of applying signifies the absence of contradictions (paradoxes), and provides for combining them in a proper sequence. Nevertheless, applications testify for existence of shortcoming in many techniques of reasoning stemming from the predicate calculus of the first order. This is known for a long time, and this is fly methods of fuzzy logic emerged together with the theories of belief and the possibilistic approaches to determine preferences.

It became clear recently, that the substantial part of failing cases of reasoning happens because the multiresolutional structure of representation is not taken in account by the process of reasoning, both in living creatures and in computer equipped intelligent systems. What is true in one level is not necessarily true in the adjacent levels. The temporal factor creates difficulties in a regular predicate calculus. Now, the situation gets aggravated by the different time scales. These considerations can be illustrated by the multiple examples. In many of them, reasoning is affected by the transformation of representation: while the quantities change the list of objects is changing, too [29], and this affects the results of generalization. It was demonstrated that the motion of "pointing" in living creatures was affected by the different time scales at the different level of abstraction in brain [30].

Finally, it has been found from many observations that the logic of natural language is different from the one presented in the theory of predicate calculus of the first order. The inferences implied by the natural language discourse to not allow to be easily transformed into statements of the predicate calculus while their implications are eventually properly interpreted and understood by humans. It is tempting to develop a) a theory of natural language reasoning and b) an automated system that would allow to use the advantages of natural language reasoning for artificial intelligent machines. The researchers of Drexel University are working on these topics now.

3.3 Resolution

The term *resolution* related to the accuracy of detail in representation and sensor output is often confused with the term resolution from the subsections of logic in artificial intelligence (resolution-refutation). Resolution of the system's level is determined by the size of the indistinguishability zone (granule) for the representation of goal. model, plan and feedback law. Any control solution alludes to the idea of resolution explicitly or implicitly.

Resolution determines the complexity of computations directly because it determines a number of information units in a representation. In complex systems and situations one level of resolution is not sufficient because the total space of interest is usually large, and the final accuracy high enough. So, if the total space of interest is represented with the highest accuracy, the ε-entropy (the measure of its complexity) of the system is very high.

The total space of interest is to be initially considered at a low resolution. Only one subset (or a limited set of subsets) of interest is further analyzed with higher resolution, and so on, until the highest resolution is achieved. This consecutive focusing of attention with narrowing the subsets' results in a multilevel task decomposition. The following terms are used with resolution intermittently: granulation, scale. "Granule" is another term of the distinguishability zone (pixel, voxel). Scale is considered to be equal to the inverted value of the granule (or an "ε-tile). When the space is intentionally discretized, we use the term tessellation, and a single granule is called "tessellatum" or "tile."

3.4 Multiresolutional Representation

The term *multiresolutional representation* is defined as a data (knowledge) system for representing the model of our system at several levels of resolution (or granulation, or scales). In order to construct a multiresolutional (multiscale, multigranular) system of representation, the process of generalization is consecutively applied to the representation of the higher levels of resolution. As a result of applying the algorithm of generalization to the modules of ELF emerge (Figure 2) with the new level of Sensory Processing (SP), World Model (WM), and Behavior Generation (BG). These new, more generalized BG-WM-BG sets are attached to the initial ELFs as the next "floor" of this structure. If further generalization is performed on the modules of the new level, an additional level of SP-WM-BG of the structure would emerge.

Multiresolutional representation can be underlaid by an ERN principle of constructing the model. Objects, relations, and actions of the ERN at the new level are different, and thus, the rules are different and the results of searching for the best course of actions are presented in different terms. However, if necessary one can substitute it by other techniques of representing experimental knowledge, e.g. by using analytical models with different accuracy of approximation.

3.5 Generalization

The term *generalization* is a formation of new entities (groups, classes, assemblies) where parts to be assembled are not prespecified. and new classes of properties can emerge. The *synonym* for the term *generalization* is (in some cases) *abstraction*, however more frequently the meaning of *generalization* exceeds the more narrow meaning of abstraction. *Antonym* - instantiation. *Generalization* usually presumes grouping (clustering) of the subsets focused upon as a result of searching and consecutive substitution of them by entities of the higher level of abstraction. This is why instead of term *resolution* levels we use sometimes an expression *levels of abstraction*, which means the same as levels of *generalization*, or levels of *granularity*. Example: In most of the cases when humans encounter new situations they face the need to create groups. They make groups or assemble together components, which are not specified as parts belonging to each other, and new classes of properties should be proposed on the flight.

From the definition of generalization, one can see that it can be performed through applying the following operators jointly: *grouping, focusing attention, combinatorial search* (or a simple *search*). There are many operators that exhibit these functions: many algorithms of clustering that can be used to perform *grouping*, many algorithm of choosing the subset of interest, e.g. windowing operators that perform focusing attention, many algorithms of search, or search equipped with combinatorial generation objects among which the search is done. To simplify further analysis of architecture we will call them operators of G, FA, CS, or about an integrated operator of GFACS.

It would be instructive to demonstrate how the term *generalization* differs from terms *aggregation* and *abstraction*. *Aggregation* is formation of an entity out of its parts. Each of the parts can be also obtained as a part of aggregation. *Synonym* - assembling. *Antonym* - decomposition. Example: The entity is formed out of its parts. Information of belonging is contained in the description of the objects. We will consider this process to be an example of a very simple group formation: we know what is the whole, and we know what are the parts. Assembling of parts into the whole, or formation of an aggregate is determined by specifications.

Formation of a class of objects which is characterized by the same property, and labeling this class with the name of this property is called *abstraction*. *Synonym* - class formation. sometimes, abstraction. *Antonym* - specialization. Example: The properties, which characterized objects can be considered objects by themselves. We won't be surprised if one calls *kindness* an entity. The fact that color is a property belonging to the most physical objects of the real world makes it an important scientific and technological entity of the system of knowledge. It is important to indicate that formation of such entity is possible only by grouping together all similar properties of different objects. A red apple, red ink, red bird, red cheeks, they all belong to the class of objects containing "redness".

So, generalization performs aggregation even when parts are not specified. This means that it subsumes the aggregation. It subsumes the abstraction, too. In all cases concerning abstraction the term generalization is applicable. Generalization is typically applied when a similarity and observations are discovered and a general rule should be introduced. The term abstraction is inappropriate in this case. Conclusion: *generalization subsumes both aggregation and abstraction*. This is a more general procedure for which aggregation and abstraction are particular cases.

3.6 Nesting

Nesting is a property of recursively applying the same procedures of multiresolutional knowledge processing by using the operator of processing at a level for consecutively processing information of all levels. The results of Sensory Processing of all levels are nested one within another, World Models are nested one within another, and the decisions generated within the module of Behavior Generation are nested one within another. Levels of a multiresolutional ELF are nested one within another, while the levels continue to function as separate independent ELFs. This separation of levels is a result of a need to reduce the complexity of computations. Thus, instead of solving in one shot the whole problem with the maximum volume of the state space and with the amount of high resolution details one may choose to solve several substantially simpler problems that are nested one within another.

3.7 Learning

The process of generalization upon the time-varying functions of a control system is called learning. As a result of this process, the statements of experiences related to elementary objects, relationships between objects, statements of actions merge together ito statements related to clusters of objects, relationships and actions. These, generalized statements allow for construction of rules. Then, all this newly obtainted set of statements can be generalized again. This process that is being performed recursively and is successively applied to its own output results in creating and constant updating of the multiresolutional system of representation, and thus, in improvement of plans and feedback control laws. Learning is a component of this multiresolutional knowledge processing. Evolution of knowledge of the system can be demonstrated as shown in Figure 6.

Obviously, learning is tightly linked with the property of Intelligent Systems of being equipped by the systems of knowledge representation (e.g. the module of World Model in ELF). This module of representation might not necessarily be physically lumped in one specific place: WM can be distributed over a multiplicity of agents, or otherwise over the physical medium used in the intelligent system.

Updating of the World Model and enhancement of its multiresolutional system of knowledge representation is done by the process of learning, which employs the set of GFACS operators that has been described above. Levels of resolution are selected to minimize the complexity of computations. Planning and determining of the beneficial feedback control laws is done also by joint using of generalization, focusing attention, and combinatorial search (GFACS).

The operation of learning was associated with layers: each layer learns separately. Learning experiences can be organized only by using a multiresolutional structure. Levels are not hard-wired, they are constructed from the information at hand. As it is done in neural net, for example. Mathematics of various operators of focusing attention, grouping and searching usually employed by GFACS algorithms can be found in [31].

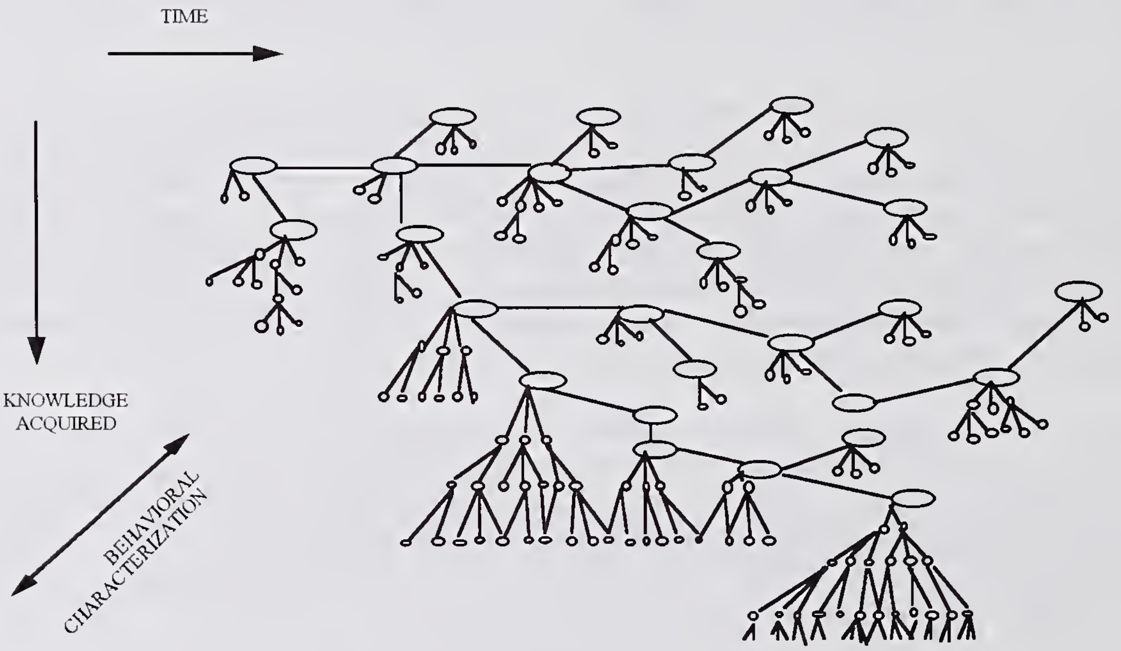


Figure 6. Evolution of World Model as a Result of Learning

One can see from Figure 7 that Learning in an intelligent system boils down to collecting experiences, applying GFACS to them, and explicating objects, actions, rules and theories that might be used by the module of Behavior Generation. Combining Figure 7 with Figure 5 gives an opportunity to learn not only from real experiences of acting within the environment but also from the imaginary experiences of simulating within the imagination of the intelligence.

3.8 Intelligent Control

Intelligent control is a computationally efficient procedure of directing to a goal of a complex system with incomplete and inadequate representation and under incomplete specification of how to do this in an uncertain environment. Intelligent control, typically, combines planning with on-line error compensation, it requires learning of both the system and the environment to be a part of the control process. Most importantly, intelligent control usually employs generalization (G), focusing attention (FA) and combinatorial search (CS) as their primary operator (GFACS) which leads to multiscale structure. In all intelligent controllers, one can easily demonstrate the presence of the GFACS operators. It also is possible to demonstrate that using the set of GFACS operators is not typical for conventional controllers, although the elements of GFACS are often utilized.

Not accidentally, at the dawn of intelligent control it was associated with using neural networks (NN), fuzzy sets (FS) and generic algorithms (GA) for control purposes. (In some publications, these tree subjects are considered a must for intelligent control). In fact, neural networks is a tool for generalization in the vicinity of the state space, fuzzy systems allow to expand the process of generalization to the larger domains of the state space. GA is just a particular case of combinatorial search with some component of internal generalization for learning purposes). In other words, NN+FS+GA is a particular case of GFACS. Thus, the views presented above are confirmed.

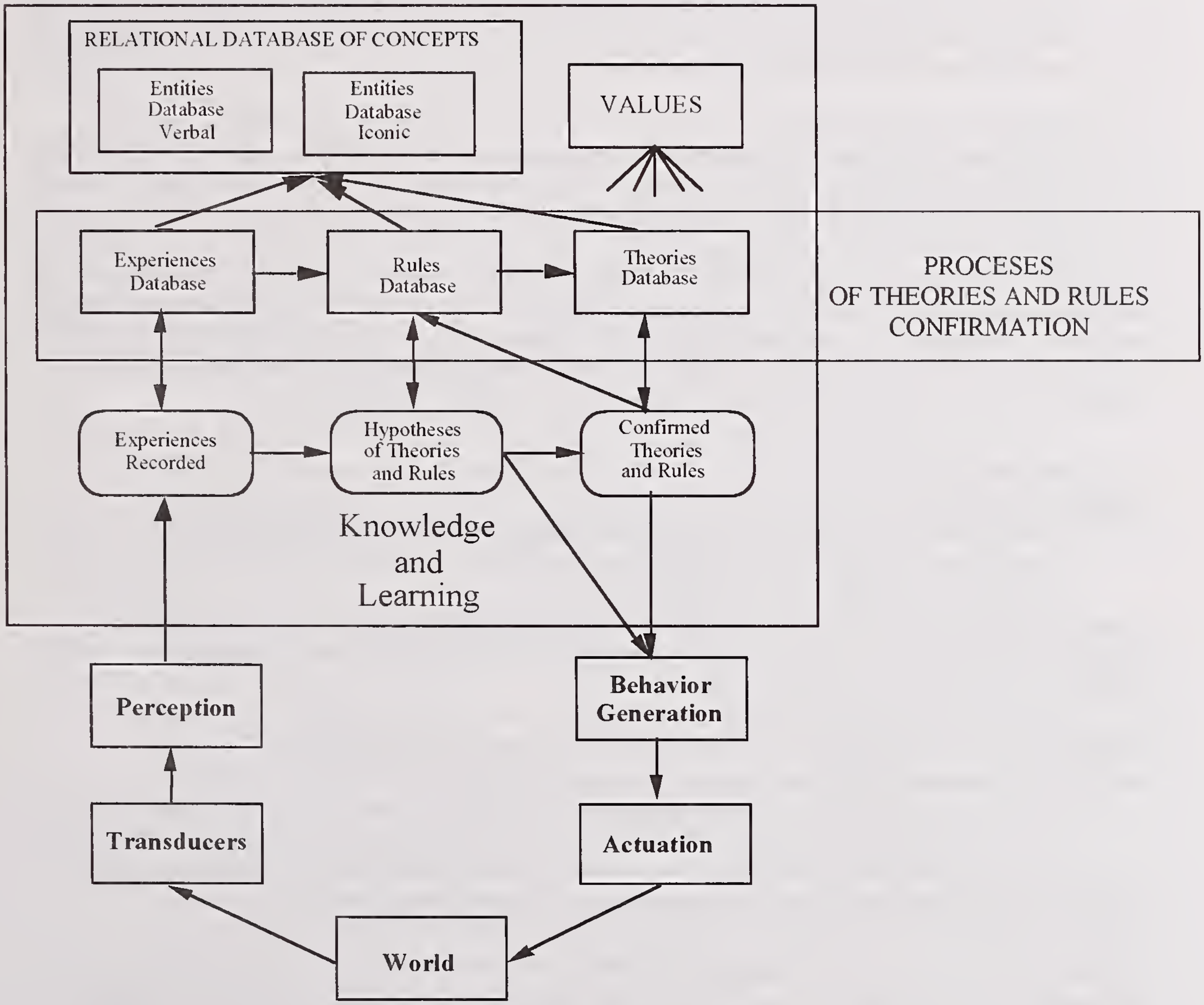


Figure 7. ELF with Learning

4. More Formal Definition of the term "Intelligence"

Most of the literature on *intelligence* can be found within the stream of publications related to psychological sciences. Most often, this is not exactly the intelligence that is discussed in this paper: they are talking about *human intelligence* primarily (like in [32]). However, even the most fundamental collections of sources do not define intelligence in the way other than listing of the mental abilities that are components of intelligence. We already spoke about immensity of *abilities* associated with intelligence. In the sources related to psychological science, intelligence is typically defined as a mental quality that combines:

1. ability to learn from experience
2. ability to adopt to new situations
3. ability to understand and handle new concepts
4. ability to acquire and use knowledge

The following definition is based on a term *thinking:*

> "[An] action exhibit intelligence, if, and only if, the agent is thinking what he is doing while he is doing it, and thinking what he is doing in such a manner that he would not do the action so well if he were not thinking what he is doing" ([1], p.29). *Thinking* is understood as a process of mediation between inner activities and external stimuli. It always alludes to the need in a specific language of thought [33] and provide for a substantive link between the mechanisms of intelligence and a computational process [34].

Several definitions are presented in [35]. One of them belongs to J. Albus and can be found in [21]:

> "An intelligent system has the ability to act appropriately in an uncertain environment, where an appropriate action is that which increase the probability of success, and success is the achievement of behavioral subgoals that support the system's ultimate goal."

This definition generalizes upon multiple abilities mentioned in the psychological definitions and introduces a concept of a success associated with the behavioral subgoals that presume some hierarchy of activities (with an inevitable multiresolutional representation). This definition is dominating since it is not linked with a particular configuration, neither it alludes to any particular domain of application. Clearly, one can apply this definition for both living creatures and artificial intelligent systems.

The operational definition introduced by A. Meystel and partially presented in [33] explains how the intelligence works:

> "Intelligence is a property of the system that emerges when the procedures of focusing attention, combinatorial search, and generalization are applied to the input information in order to produce the process of intelligent system functioning."

Focus in this definition is how information is processed so that it makes this mechanism *intelligence*. Earlier in this paper, there was more about procedures involved (GFACS) and representations required (World Model in one of the available forms of e.g. ERN).

More technologically inclined definition from [33] demands to concentrate on the issue of uncertainty (that was already mentioned in Albus' definition):

> "Machine intelligence is the process of analyzing, organizing and converting data into knowledge, where machine knowledge is defined to be the structured information acquired and applied to remove ignorance or uncertainty about a specific task pertaining to the intelligence machine."

Focus in this definition: converting data into knowledge by removal of uncertainty.

All the above definitions can be supplemented by informative statements describing features typical for intelligence but not reflected in the definitions. For example, a technological system with intelligence, i. e. intelligent system, undoubtedly can deal with unanticipated factors due to the ability to learn:

> An intelligent system must be highly adaptable to significant unanticipated changes, and so learning is essential. It must exhibit a high degree of autonomy in dealing with changes. It must be able to deal with significant complexity, and this leads to certain sparse types of functional architectures such as hierarchies.

As a byproduct of all these abilities, a number of additional features emerge gradually in a developing intelligent system. For example, a feature of *autonomy* is associated with intelligence although we still do not know how. Dealing with complexity requires using multiple resolutions, because functional hierarchical architectures are declared. Thus, the feature of being based upon multiresolutional representations is a fundamental one. Having in its core a semiotic closure is typical for an intelligent system, too.

Now we try to create a synthetic definition absorbing the definitions and supplementary statements above:

> Intelligence is a control tool that has emerged as a result of evolution by rewarding systems with increase of the probability of success under informational uncertainty. Intelligence allows for a redundancy in its features of functioning simultaneously with reduction of computational complexity by using a loop of semiotic closure equipped by a mechanism of generalization for the purposes of learning. Intelligence grows through the generation of multiresolutional system of knowledge representation and processing.

The multi-level systems fitting into this definition are not necessarily hardwired hierarchies. They are *virtual hierarchies* of perception, of knowledge- representation about the world model, and of decisions about behavior generation. As a new concept of "knowledge" emerges, a new "node" of the representation ERN is being created.

From this effort to scan existing and create a new definition, new analytical and research tasks precipitate. It becomes clear that the system of intelligence should be equipped by a capability to properly measure the objects, relations, actions and behaviors. Thus, the problem of evaluating metrics of performance and intelligence emerges. It becomes clear that intelligence can be evaluated by "a degree of intelligence". One can see that the definitions explored in this sub-section allude to the need of measure and perform quantitative ranking that is supposed to end up with a choice of a decision making. The definition of intelligent control should be based on the properties of intelligence as we understand them rather than the virtue of using some particular hardware components.

There is the list of factors that are supposed to be measured for evaluating an intelligent system:

- *Complexity Reduction.*
  Complexity should be evaluated and possibly the lowest level of complexity should be preferred under other similar conditions. Reduction of complexity should not bind the capability to develop redundancy.
- Redundancy
  A measure of "exceeding" the immediate needs should be injtroduced; one can see that the ability to evaluate the "immediate needs" is required.
- *Increase in Functionality*
  The design specifications can be used to evaluate the measure for both "immediate needs" and items of "functionality." However, in many cases, the specifications are not available. Definitely, an ability to evaluate functionality quantitatively would be an advantage, but we have to know how to restore their list.
- *Multi-level Systems.*
  The practice of intelligent system design demonstrates that the number of resolution levels is being selected based upon heuristics, not clear mathematical analysis of advantages and disadvantages this number entails. Designers do not know how many levels should a system have, and what are the other quantitative factors involved in assigning a number of levels to a multiresolutional system.
- *Degree of Intelligence.*
  A measure of intelligence is presumed to be known, at least a relative measure (which one system is smarter if general parameters are the same but different mechanisms of sensory processing, or different algorithms of planning are applied).
- *Degree of Autonomy.*
  A measure of Autonomy is presumed for the systems that are supposed to decide their own course of actions for themselves.
- *GFACS*

At the present time there is a multiplicity of the mechanisms (algorithms) of generalization. We do not have any basis for comparing the results of their functioning. Even in more simple cases (e.g. focusing attention in ARMA algorithms we do not have recommendation of comparing different versions of ARMA.

- *Increase in probability of success.*

  How should success be evaluated depends on our ability to specify it. (Is this money, power, knowledge, ability to live longer? Are these outcomes anticipated as a measure of success when they are computed for the system under consideration, or for the group, or for several generations?)

5. Evolution of Intelligence

In nature, the evolution of intelligence can be demonstrated as the development of a tool of survival. This tool evolved in living creatures (systems) as a control mechanism (a controller) to optimize needs satisfaction in changing environment. As the complexity of needs was growing, in addition to creating ways of their satisfaction the duty was performed to accordingly develop the mechanism of intelligence. The major destination of intelligence is to solving harmoniously the combined task of *NEEDS SATISFACTION + COMPUTATIONAL COMPLEXITY REDUCTION.*

Increasing functionality for performing this task can and should be measured. The evolution of intelligence presumes the evolution of both the system and the controller. The proper measure allows to judge results of evolution of intelligence. Evolution or development allows for increasing the functionality of the system jointly with reduction of its computational complexity. This is why the ability to *generalize* emerges, as the ability to lump entities of matter and/or information for more effective storing and computation. Generalization is a tool of creating new, abridged systems and their representations. It is a tool of creating representations in generalities, creating new levels (generalized)of lower resolution with new metrics or granulation. At the lower level of resolution, the tools of intelligence can afford a larger scope of attention, solve a problem of a larger picture, with a longer horizon of planning. So, the decision making on any given resolution should be preceded by the preplanning at a lower resolution level.

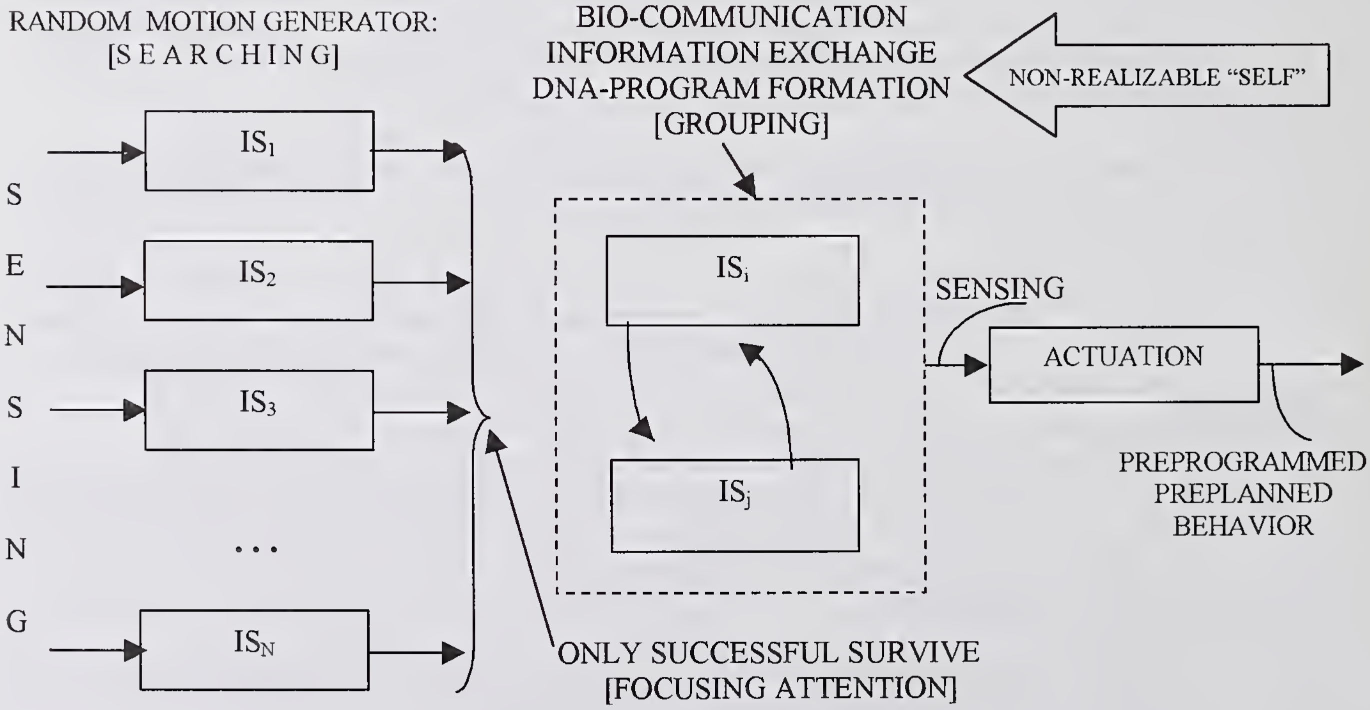


Figure 8. Architecture of E.coli's Intelligence

The biological models allow us to observe the growth of the degree of intelligence in the living forms starting with single cell organisms, through E.coli [6], via substantially more complicated living forms from mollusks to mammals [36]. and concluding with a human being [32] (See Figures 8,9,10 developed for E.coli level within the paradigm of [6] with [37]).

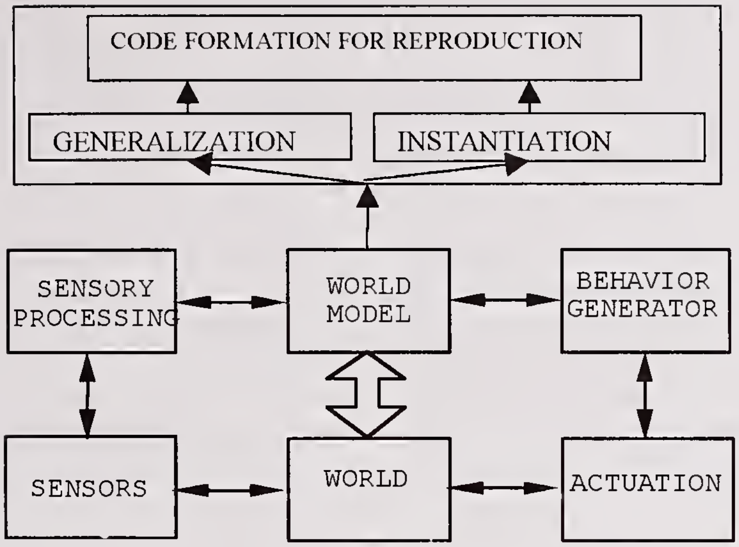


Figure 9. Developing World Model via reproduction

Figure 8 shows the group level of the E.coli intelligence architecture. Each individual E.coli is shown as a separate intelligent system IS$_i$. Random motion of all E.coli in this group (combinatorial search) leads to the situation that only those moving successfully survive (focusing attention). The survived E.coli individuals communicate and share their experience via reproduction (grouping). The successful behavior is s a result of this bio-information exchange and the results of GFACS operations are stored in the DNA, and the group (as a level) changes its behavior, and produces a behavior which increases the fraction of survived individual in the group. Thus, it can be considered "preprogrammed, preplanned behavior" at the level of the group. The group ELF has actually as modified its World Model as shown in Figure 9.

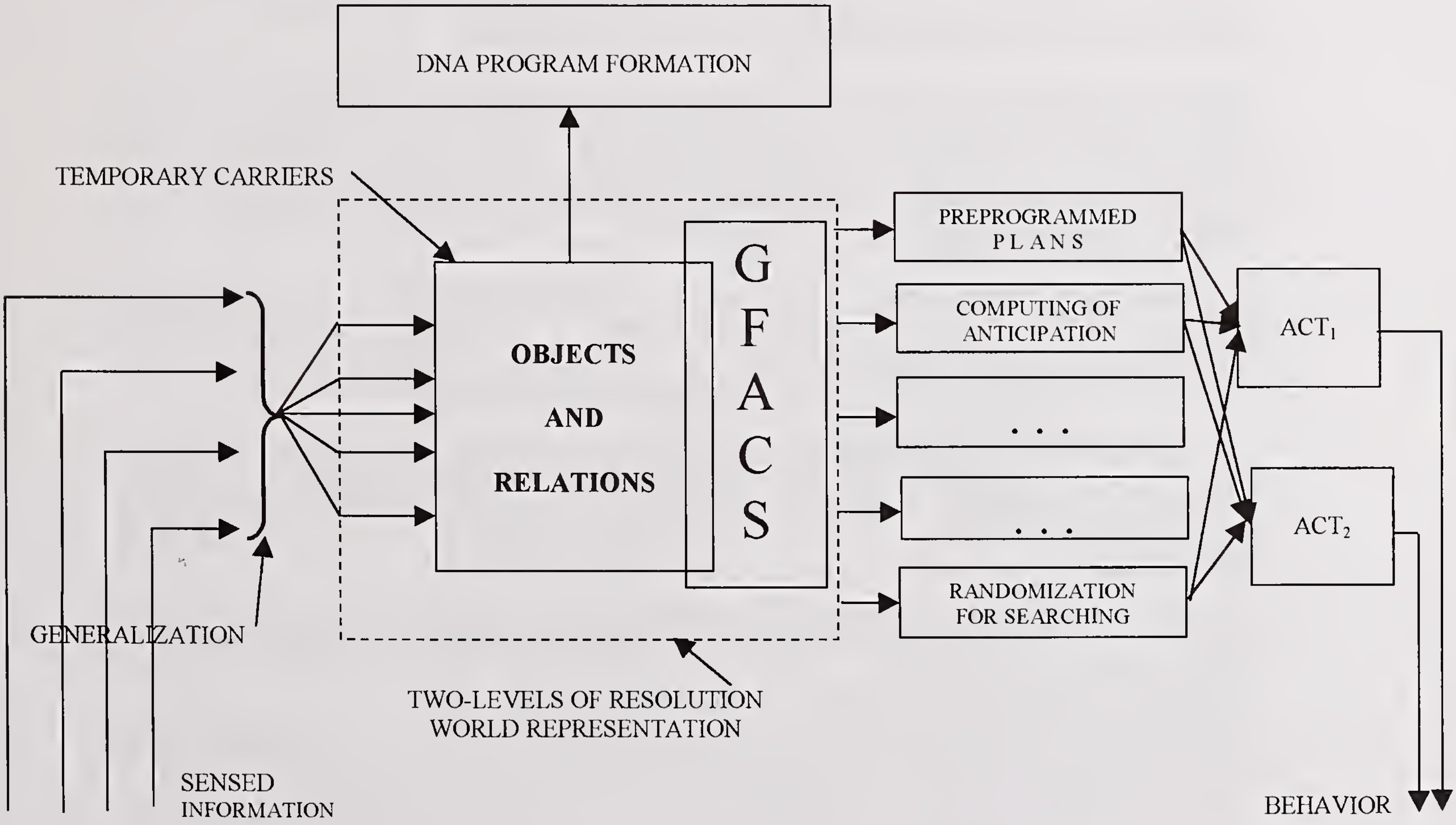


Figure. 10 Evolution from group intelligence to individual intelligence

The architecture shown in Figure 8 can be developed into structure presented by Fig. 10, where learning is performed not through sacrificing unsuccessful individual but through abandoning unsuccessful "theories" tested by internal simulation within the rudimentary just emerging system of representation. Here, instead of the multiplicity of individuals we have ERN of objects and relations that are generalized not by communication via reproduction but computationally by applying GFACS embodied as a set of information processing procedures.

Similar evolution can be observed in the domain of technology and in the domain of Linguistics. The processes of intelligence evolution extracted from these domains can be discussed in a generalized form by using architectures similar to Figures 8 through 10. Some of the advanced technological architectures (e.g. RCS) are described in [38].

Interesting temporal effect can be anticipated in the process of evolution. One can easily anticipate that the evolution is a "punctuated" one[2] (in the sense of [36]) since the new blocks only occasionally emerge in the architectures of intelligent systems.

6. Mechanisms of Intelligence

Analyses of the processes of structural evolution in the area of intelligence allow for discovering the following mechanisms of intelligence.

- A semiotic closure is the basic structure of intelligence (see Figure 2). It differs from a simple feedback loop because each element of the closure is a source of redundancy and a generator of the adjacent resolution levels by the virtue of GFACS operation.

- Evolution of multiple-choice preprogrammed behavior into a multiple alternative creation ends up with multiple theories development (the latter is performed in the *imagination*).

- Through combinatorial search, focusing attention and grouping performed in Nature by the mechanism of natural selection[3] the discovery of more efficient techniques was done. It was discovered by the intelligent agents that storing information about objects of the world, actions they encounter, and rules entailed by the changes is more efficient. Indeed, it is less expensive than testing the same material (often, living) samples again and again to receive similar results.

- Generalization and learning through natural selection from the choices created by material alternatives has demonstrated to be a waste of time, energy and matter. It is more efficient to learn by dealing with information only, i.e. by theorizing (THEORY → THE RESULT OF GENERALIZATION UPON RULES, RULE —>RESULT OF GENERALIZATION UPON EXPERIENCES)

- Mechanisms of generalization give a consistent explanation to the semiotic tools of evolution discovered earlier in [39, 40]. The same mechanisms and tools determine that the ultimate methodology of analysis of the mechanisms of intelligence can be defined and realized successfully in the domain of Natural Language analysis.

- Any RULE discovered by an intelligence is a statement of some generality: it cannot refer to all details of realistic test cases. The selection of the proper details for the particular state of affair is performed by the mechanism of focusing attention.

- Multiresolutional storage obtained via consecutive generalization turned out to be the most efficient method of storing information.

[2] Punctuated evolution demonstrate periods of changes with intervals of the absence of any development.

[3] Actually, the reader should have already anticipated the conjecture that natural selection in the Nature played a role similar to the algorithmic mechanisms of generalization and learning.

7. Intelligence of the Conventional Controller

It would be desirable to determine what is the relation between the conventional control and intelligent control. The following statements are based on the preceding materials.

1. Conventional control is about feedback. The goal formation is external to the problem. When we include the goal formation the problem become IC-embedded because the goal for each level of the higher resolution is created as a result of BG-module functioning at the level of lower resolution.

2. The structures of intelligent control are formed as semiotic closures, mostly the multiresolutional ones, which contain an element that can be called "feedback". But feedback is not the entire issue. The transformations within the feedback loop are more important. The classical feedback does not need to have any redundancy in it. This is why Y.-C. Fu associated the concept of Intelligent Control with "recognition" in the loop.

3. We would expect that the feedback of the semiotic closure contains GFACS as a rule.

4. Optimization as a part of functioning of the conventional controller presumes searching at a level but stops short from recognition its embedding within the multiresolutional hierarchy of top-down constraint propagation.

References

1. G. Rile, *The Concept of Mind*, Hutchinson, London, 1949
2. A. Meystel, "Intelligent Systems: A Semiotic Perspective", *International Journal of Intelligent Control and Systems"*, Vol.1, No. 1, pp. 31-58
3. Y. Maximov, A. Meystel, "Optimum Architectures for Multiresolutional Control", Proc. IEEE Conference on Aerospace Systems, May 25-27, Westlake Village, CA 1993
4. A. Meystel, "Planning in a hierarchical nested controller for autonomous robots," Proc. IEEE 25th Conf. on Decision and Control, Athens, Greece, pp. 1237-1249
5. Eds. H. von Foerster, G. W. Zopf, *Principles of Self-Organization*, Transactions of the Symposium on Self-Organization, U. of Illinois, June 8-9, 1961, Pergamon Press, Oxford, 1962
6. K. M. Passino, "Distributed Optimization and Control Using Only a Germ of Intelligence", Proc. of the 2000 IEEE Int'l Symposium on Intelligent Control, Eds. P. Groumpos, N. Koussoulas, M. Polycarpou, Patras, Greece, 2000, pp. P5-P13
7. J. S. Parkinson, D. F. Blair, "Does E.coli Have a Nose?" *Science*, Vol. 259, 19 March 1993, pp. 1701-1702
8. A. N. Kolmogorov, "On Some Asymptotic Characteristics of Bounded Metric Spaces," Proceedings of Academy of Sciences, Vol. 108, No. 3, 1956 (Doklady Akademii Nauk, in Russian).
9. M. Rackovic, D. Surla, M. Vukobratovic, "On Reducing Numerical Complexity of Complex Robot Dynamics," J. of Intelligent and Robotic Systems, Vol. 24, 1999, pp. 269-293
10. E. Weyuker, "Evaluating Software Complexity Measures," IEEE Transactions on Software Engineering, Vol. 14, No. 9, 1988, pp. 1357-1365
11. 11. D. Boekee, R. Kraak, E. Backer, "On Complexity and Syntactic Information," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-12, No. 1, 1982, pp. 71-79
12. Y. Abu-Mostafa, "The Complexity of Information Extraction," IEEE Transactions on Information Theory, Vol. IT-32, No. 4, 1986, pp. 513-531
13. G. Zames, "On the Metric Complexity of Causal Linear Systems: Epsilon-Entropy and Epsilon-Dimension for Continuous Time," IEEE Transactions on Automatic Control, Vol. AC-24, No. 2, 1979, pp. 222-230
14. A. Meystel, "Architectures, Representations, and Algorithms for Intelligent Control of Robots," (Chapter 27) in Intelligent Control Systems, eds. by M. Gupta and N. Singha, IEEE Press, 1995, pp. 732-788
15. A. Meystel, "Multiresolutional Architectures for Autonomous Systems with Incomplete and Inadequate Knowledge Representation", in Artificial Intelligence in Industrial Decision Making,

Control, and Automation, eds. S. G. Tzafestas, H. B. Verbruggen, Kluwer Academic, 1995, pp. 159-223

16. A. Meystel, "Learning Algorithms Generating Multigranular Hierarchies," DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 37, 1997, American Mathematical Society, pp. 357-383
17. A. Meystel, E. Messina, "The Challenge of Intelligent Systems," Proc. of the 2000 IEEE Int'l Symposium on Intelligent Control, Eds. P. Groumpos, N. Koussoulas, M. Polycarpou, Patras, Greece, 2000, pp. 211-216
18. L. von Bertalanfy, Robots Men and Minds, Braziler, New York, 1967 [see p. 69]
19. H. Pattee, "Physical basis and origin of control," in Hierarchy Theory, Ed. H. Pattee, Braziler, New York, 1973 [see p. 94]
20. A. Meystel, "Theoretical Foundations of Planning and Navigation for Autonomous Robots," *International J. of Intelligent Systems*, Vol. II, 1987, pp. 73-128
21. J. Albus, Outline of the Theory of Intelligence, *IEEE Transactions on Systems, Man, and Cybernetics*, 1991
22. I. Rock, S. Palmer, "The Legacy of Gestalt Psychology," *Scientific American*, December 1990, pp. 84-90
23. J. McCarthy, "Generality in Artificial Intelligence," *Communications of the ACM*, Vol. 30, No. 12, December 1987, pp. 1030-1035
24. B. Porat, B. Friedlander, ARMA Special Estimation of Time Series with Missing Observations, *IEEE Transactions on Information Theory*, Vol. IT-30, No. 6, 1984, pp. 823-831
25. D. N. Politis, ARMA Models, "Prewhitening and Minimum Cross Entropy," *IEEE Transactions on Signal Processing*, Vol. 41, No. 2, 1993, pp. 781-787
26. P. Combettes, J.-C. Pesquet, "Convex Multiresolution Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 12, 1998, pp. 1308-1318
27. J. A. Bangham, P. Chardaire, C. J. Pye, P. D. Ling, "Multiscale Nonlinear Decomposition: The Sieve Decomposition Theorem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 5, 1996, pp. 529-539
28. A. Meystel, "Baby-robot: On the analysis of cognitive controllers for robotics," Proceedings of the IEEE Int'l Conference on Man & Cybernetics, Tuscon, AZ, Nov. 11-15, 1985, pp. 327-222
29. S. Murthy, "Qualitative Reasoning at Multiple Resolutions," Proceedings of the 7th Nat'l Conference on Artificial Intelligence, AAAI-88, Vol. 1, 1988, pp. 296-300
30. D. Ballard, M. Hayhoe, P. Pook, R. Rao, "Deictic Codes for the Embodiment of Cognition," BBS, Cambridge University Press, 1996
31. M. Vidyasagar, A Theory of Learning and Generalization, Springer, London, 1997
32. Handbook of Human Intelligence, Ed. R. J. Sternberg, Cambridge University Press, Cambridge, UK, 1982
33. J. A. Fodor, "Why There Still Has to Be a Language of Thought," in Psychosemantics by J. A. Fodor, MIT Press, 1987, pp. 135-167
34. G.Fauconnier, Mapping in Thought and Language, Cambridge University Press, Cambridge, UK, 1997
35. P. Antsaklis, "Defining Intelligent Control: Report of the Task Force on Intelligent Control," *IEEE Control Systems Magazine*, June 1994, pp. 4, 5, 58-66
36. S. Gould, "Triumph of the Root-Heades," Natural History, No. 1, 1996, pp. 10-14
37. M. Lieber, "Adaptive Mutations and Biological Evolution," Frontier Perspectives, Temple University, Spring-Summer 1991, pp. 23 and 26
38. J. Albus, A. Meystel, "A Reference Model Architecture for Design and Implementation of Intelligent Control in Large and Complex Systems", *International Journal of Intelligent Control and Systems"*, Vol.1, No. 1, pp. 15-30
39. G. M. Tomkins, "The Metabolic Code," *Science*, Vol. 189, 5 September 1975, pp. 760-764
40. K. L. Bellman, L. J. Goldberg, "Common Origin of Linguistic and Movement Abilities," American Psychological Society, 1984, pp. R915-R921

Machine IQ with Stable Cybernetic Learning with and without teacher

Harold Szu, Ph.D. , Fellow IEEE
Digital Media Lab, ECE Dept. GWU, 22 & H St. NW, Wash DC 20052

1. MACHINE IQ

Lotfi Zadeh has raised an interesting and philosophical question: what is the Machine Intelligent Quotient (IQ) needed for intelligent household consumer electronics and robots?

We wish to suggest a nonlinear but monotonic scale similar to the logarithmic scale adopted by C. Shannon information theorem based on the logarithmic phase space in L. Boltzmann entropy notion in Sect. 5. The justification is that the human-like creativity is rare and difficult and must be reached at the top 50% scale with unsupervised learning in Sect. 2 & 3, and the dumber machine near the low end of the scale. In between they are separated by dyadic basis. However, for household convenience, after taking the logarithmic nonlinear scale of human-like intelligence, the net result is further measured by taking the usual linear percentage scale. We concede that these double scales may be sensible in the operational definition but could not be fundamental. For we are not absolutely certain about what are the necessary and sufficient ingredients for a humankind or machine to be intelligent.

(1) After we have taken the logarithmic scale, then MIQ=10% of human being is loyal to human master and its own survivability, say, the robot having MIQ=10% is able to find and differentiate the electric power plugs having two porn's of 110 Volts or three porn's of 200 Volts.

(2) Then, MIQ=20% is able to understanding human conversation.

(3) In that direction we can extrapolate MIQ=30% to be able to read facial expression and voice tone for the emotional IQ to understanding irrational emotion need of human being.

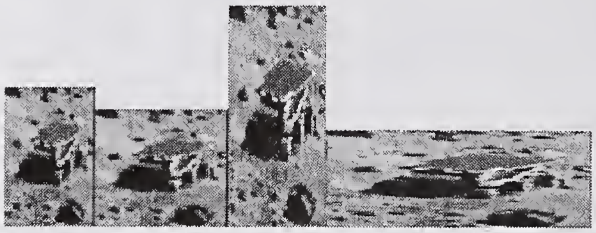
(4) MIQ=40% is able to command and control a team of other robots.

(5) MIQ=50% is able to "explore the tolerance of imprecision," e.g. using fuzzy logic to negotiate a single precision path finding in an open save terran.

We divide MIQ into the supervised learning with an open lookup table having the extrapolation and interpolation capability up to MIQ $\subseteq$ 50%. The key of human-like sensor systems is learning without supervision to be scored MIQ beyond 50%. Such a learning methodology is necessary, because, other than factory robots, any indispensable need of robots happens usually in an open, uncooperative and hazardous environment with an unforeseeable nonlinear dynamics interwoven with non-stationary complexity. We believe that unsupervised learning is necessary in building of human-like trial-and-error estimation systems, such that a major next step toward the intelligent robot is visual and natural language understanding needed for self-determination in uncontrollable environment. Thus, laboratory simulations of robotic teams are necessary with the help of the wireless video feedback control technology base (WaveNet video communication devices on wheels).

Intelligent processing is a need common to real world surveillance and control, especially in unmanned environment, namely the nucleus reactor chamber, the undersea, and the outer space. To set the research direction of real world applications, we consider a not-yet solved and perhaps a milestone problem to design an intelligent robot entering into a new challenging situation (not unlike a newborn infant facing a bustling and hustling world). This is challenging because the robot has not yet acquired any pertinent internal knowledge representation. The robot must learn how to process the unfamiliar sensory, hearing and vision inputs and to travel through an uncharted environment (even if the robot has been endowed with the past experience and has had a man in the loop as the coach for the remote guidance and control). In a distant and novel situation, we anticipate the robot having some difficult in following the supervised learning given that the robot has not yet learned what is the desired output for the unfamiliar inputs. The milestone problem is thus due to the impossibility to specify all details ahead of the time, and therefore it is important to develop unsupervised sensory learning capability (which leverages subsequently the self-supervised learning with the gradually acquired experience).

A real world challenge happened recently during the NASA PathFinder exploring the Mar, of which the round trip time for Control, Command, & Communication (C3) is 5 minutes. A futuristic suggestion would be using ANN for intelligent pathfinder leader to control the rest of team members, and then local control will be instantaneous, while we control the leader with non-real time commands (not unlike biologically a queen bee leading a hundred working bees).



Martian pathfinders would require a local adaptive and intelligent control because of time-delay of the ground station. Thus, instead of one-to-one, we have proposed a quarterback robot (25% IQ) controlling a team of linemen robots (not unlike a queen bee controlling a hundred working bees). A team of Unmanned Marine Vehicles, Dolphins, can parallel hunt for mines in the extended littoral battle space. Using wireless video feedback control (WaveNet via SINCGARS), a person as the coach may communicate with some delay (because of out of the line of sight) the goal with an intelligent robotic (quarterback of the dolphins team). The quarterback is able to execute locally and faster C3 of all linemcn UMV's to identify objects in voting and going around local obstacles, using the prior GPS information from the wide receiver scoutimng UMV.

2. LEARNING METHODOLOGY

Recently, Irwin has edited the Industrial Electronics (IE) Handbook (CRC & IEEE Press, pp.1-1686, 1997) and devoted one thousand pages to the intelligent electronics (IE) describing comprehensively all enabling technologies. These are expert systems and neural networks, fuzzy systems and soft computing, evolution systems, computational intelligencc, and hybrid applications, and emergent technologies. We believe that some degree of human-like intelligence is necessary for user-friendly interaction with IE. On the other hand, the classical artificial neural networks (ANN) with *supervised* learning strategy have reached the maturity and plateau with some mixed appraisals, although the interdisciplinary studies have just been bearing fruits (since the establishment of international neural network society a decade ago). For example, the neural physiological experiments of human sensors have culminated a truly *unsupervised* learning new paradigm. When a newborn baby faces the bustling and hustling world, he/she cannot grasp the changing signals $\mathbf{s}_i(t) \equiv s_i(t)\mathbf{a}_i$, from noisy inputs $x_1(t)$, $x_2(t)$. However, the intuition is that non-noises must be signals. Thus, the child recognizes the fluctuating noise that has zero correlation $<v_1(t)v_2(t)>_G = 0$ of the neuron outputs.

If one assumes a linear instantaneous inputs as

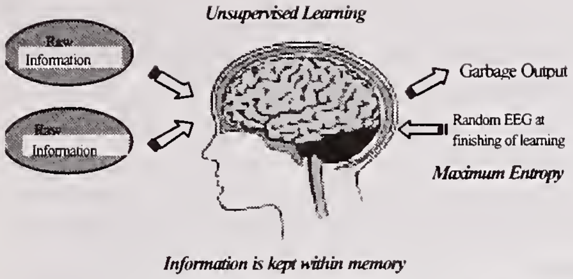
$$\mathbf{X}(t)=s_1(t)\mathbf{a}_1+s_2(t)\mathbf{a}_1\equiv[\mathbf{A}]\mathbf{S}(t) \qquad (1)$$

where $\mathbf{S}(t)$ and $[\mathbf{a}_1, \mathbf{a}_2] \equiv [\mathbf{A}]$ are unknowns, then the artificial neural network (ANN) seeks a weighted sum: $\mathbf{V}(t) \approx [\mathbf{W}]\mathbf{X}(t)$ which will produce garbage outputs defined by

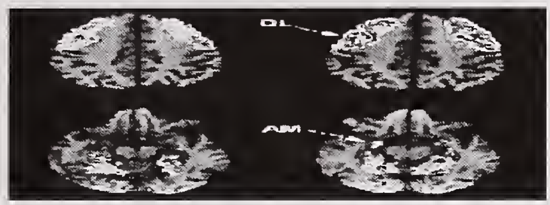
$$<\mathbf{V}(t)\mathbf{V}(t)^T>_G = [I]\approx[\mathbf{W}][\mathbf{A}]<\mathbf{S}(t)\mathbf{S}^T(t)>[\mathbf{A}]^T[\mathbf{W}]^T$$

which implies $$[\mathbf{W}][\mathbf{A}] \approx [I] \qquad (2)$$

because $<\mathbf{S}(t)\ \mathbf{S}^T(t)> \approx [I]$ has a nontrivial higher order statistics (HOS). Thus, the internal knowledge $[\mathbf{W}]$ is discovered as $[\mathbf{A}]^{-1}$. This math is called the Independent (i.e. joint probability density factorization) Component Analyses (ICA). For instance, after the external stimulus by light, sound, and perhaps touch sensations, one hundred millions of visual, hearing and tactile sensory neurons generate highly redundant collective excitations, which can not and should not sustain themselves. Local time scale complex nonlinear dynamics will always yield to decaying in the global time scale, according to Neurodynamics Lyaponov-like Theorem, proved in Sect. 6. Unsupervised learning takes the advantage of the necessary decay of those highly redundant excitations, as the mean of memory toward statistically independent components (IC) without knowing precisely what they are. Therefore, this output state can not be specified ahead of the learning in the truly unsupervised fashion. After a decade studies of neural nets, we have realized that the chief biological reason for a pair of sensors, eyes, ears, tongue sides, nose holes, hands, is to provide the robustness redundancy and the instantaneous spatial temporal de-noise without teacher together with a simultaneous recognition with teacher (associative memory). To simplifying the unsupervised portion of learning, two ears disagree must not be signal--a perfect de-noise algorithm (called mathematically Independent Component Analyses (ICA)), i.e. "pair of raw inputs, garbage output" as opposed to a dumb PC: "garbage in, garbage out". Since the output is garbage, no teacher is needed, and what's kept inside the brain without being squeezed out as noise is useful feature **Fig. 1.** Neuro Paradigm



Brain imaging experiments might support the hypothesis that learning of a newborn child might have various stages interwoven at ease. Advanced brain imaging, e.g. Functional Magnetic Resonance lmaging (fMRl) or Positron Emission Tomography (PET), can help substantiate the unsupervised learning. For example, Positron Emission Tomography (PET) image reveals all female subjects using both side of brains while all male subjects use only one-side to processing speech. That radioactive-labeled glucose supplies the nutrition needed in the brain processing can decay position rays, which, furthermore, decay into two opposite Gamma rays onto all around films. lmage is synthesized like the Tomography, but the PET radiation comes from selectively inside rather than indiscriminately in traditional Tomography.



Should the collective neural activity randomizes into the de-correlation, as stated, the intensity of positions and associated 2 gamma rays that have collectively lit up the brain imaging would fade away as noisy-like. To carry the thought experiment further, we consider instead adults an infant just born. We conjecture that, without yet any internal knowledge representation, a newborn baby, who does not have anything pain/pleasure/movement etc. to be associated with, must utilize sensory input de-correlation process toward noisy output, as a truly unsupervised learning strategy to build up the internal knowledge representation. Let us imagine that the first sight of a face of mother composed of several millions of collective excitations in human visual systems (6.1 millions color perception cones and 150 millions dark light rods on retina). This first impression must be decaying toward randomizing responses as lacking of any imagination ability of association the infant can not sustain another concept or image as the only reference point for feature extraction stoppage criterion. This innate ability could be the foundation for any artificial endowment of machine IQ. Afterwards, the traditional ANN supervised learning can be leveraged by the existence of internal knowledge representation. In this sense, the supervised learning may be called the second stage of learning. This unsupervised learning ability is amble demonstrated in blind de-mixing acoustic signals and images without knowing what the original sources are, and how they are mixed as an infant in a cradle.. This trait is mathematically referred to be lCA or BSS. The result is similar to the cocktail party effect that one can de-mix, during a noisy drinking party, the signals and detect one's name or other important messages among cross talks. There are three stages of learning as a new borne baby develops. The initial stage is related to the decay of collective excitation generated by millions of sensors (6.2 millions color perception cones and 150 millions gray-scale rods) connected to millions of neurons in the cortex area.

1. lnitial Stage no details (< 1 week/month old) The first stage happens at the limiting case of eye-opening first sight without knowing the desired association memory and without acquiring yet any internal knowledge representation. This is characterized by the merely decay of coll^ctive sensory excitations to noise. "Mar, Bar, Dog, Cat, pleasure & pain, etc." learned without feature definition not yet having meaning of association; "(sensors) info inputs = garbage output (randomized EEG when CNS learning stopped)", namely the unsupervised learning by maximizing the output Entropy. (as if lemonade has been squeezed and kept in the synaptic junctions and the useless skin & junk is throw out).
2. The second stage is efficient by leveraging the lC knowledge as the desired output lntermediate State supervised learning sensory inputs--desired outputs forming associated pairs.
3. The third stage may be called the creative sate of thinking (< K12). The third stage takes the advantage of both the internal knowledge being internally generated as realistic excitations, called active imagination, together with the externally generated sensory excitation (to seek again by the unsupervised as new knowledge base). The external sensory inputs plus internal imagination inputs generate new thought.

These stages are alternatively going on effortlessly, and may become not separable subsequently. However, this initial condition of learning should help robots with intelligent sensory processing capability to deal with new environment. ln real world applications, we often lack the precise knowledge of the desired output features. Therefore, we cannot apply the supervised ANN to associate the input data to the output feature. However, without knowing the desired output, the new unsupervised ANN can extract the independent components of the input data, which itself becomes the desired feature.

In summary, when a raw information input through pairs of eyes, ears, nasal passages, after been sieving through the synaptic weight matrix for extracting IC features, the final neural outputs become noise-like. The ANN unsupervised learning changes the ANN weight matrix to sieve or squeeze anything useful (higher order correlation information) from the input sequence until the outputs are left with (nothing but maximum entropy) redundant garbage or noise. This strategy is on the contrary to the supervised one because in a truly unsupervised learning we cannot assume any output goal but the garbage-output for information-input. In paraphrase, after squeezing the juicy clean, ANN throws away the trash. It does not matter whether the input is an orange or a lemon, the end product of the unsupervised process is identical, being without a teacher the only logical choice is trash output meaning no more useful covariance information, i.e. the unique noise-like output state. While a traditional computer has a motto to describe a dumb and do-nothing computers as "garbage-in, garbage-out", we dramatize that a smart neurocomputer has a motto that "raw information-in, garbage-out". Such a novel "information-input and garbage/noise output" paradigm for the neurocomputer has accomplished a machine I/Q, which is surely higher than a traditional computer with the do-nothing motto. The new generation or 6th gen neurocomputer can at least do a sensor feature extraction job without supervision.

2. UNSUPERVISED LEARNING

We present simple mathematical models of unsupervised learning algorithms of artificial neural networks (ANN) as motivated by the biological principle of redundancy reduction (Barrow, 1953) via statistical decorrelation of sensory mixtures, known in ANN as Blind Source Separation (BSS) or Independent Component Analysis (ICA). Different stages of learning are conjectured which could help robots acquire additional knowledge needed in a hazardous and new environment. We begin with the familiar formalism of auto-regression (AR) which is easily generalized to the supervised back-error-propagation ANN, and then to the unsupervised sensory mixture decorrelation. This is initially based on higher orders of statistics, e.g. 4th order cumulant-Kurtosis, that is, furthermore, led to the maximum entropy of all cumulants. These generalizations have been illustrated with computer simulations in a controlled setup. Real world applications are given in Part II, such as remote sensing subpixel composition, voice-dictation phoneme segmentation by means of ICA de-hyphenation, and cable TV bandwidth enhancement by simultaneously mixing all Sport and movie entertainment events.

Such a truly unsupervised learning strategy in terms of ANN is mathematically elucidated in terms of a pair of sensory inputs vector **X**(t). Assume a linear piecewise-stationary mixture model. The unknown but piecewise time-independent feature matrix [A] consists of column feature vectors $[\mathbf{a}_1,\mathbf{a}_2]$ and the correspondingly unknown vector source S(t) corresponds to the percentage of feature vector composition, then

$$\mathbf{X}(t) = [A]\, \mathbf{S}(t) \qquad (1)$$

Through ANN feedback iteration unsupervised learning of the synaptic weight matrix [W]. The bipolar unitary output may be approximated at the maximum entropy (cf. Appendix A Proof)

$$\mathbf{V}(t)= \tanh([W]\mathbf{X}(t)) \approx [W]\mathbf{X}(t), \qquad (2)$$

which has a linear slopc at the threshold value of input for "may-be-yes may-be-no" no-information answer (proved to be at the maximum Shannon entropy with the minimum of information; rather than, the definite yes-or-no informative answers at the large input bipolar values). Instead of tanh one can also adopt the positive sigmoid function, whose outputs must be subtracted by the averaged value to be bipolar fluctuations of mean-zero. In any case, the output is not the traditional desired output, but must be reduced to be noise-like at the end of unsupervised learning process in consistent with the maximum entropy with no more output information at the second moment level. The off-diagonal random components on the time average vanish, while the diagonal elements are identically squared and never vanish and can be normalized to one.

$$<\mathbf{V}(t)\mathbf{V}^T(t)>= [I] \qquad (3)$$

To achieve it, a random permutation of a large block of data is often recommended to avoid the sampling inaccuracy. (The time order scrambled for fear of sampling error will be preserved in data **X**(t), once we have found [W] as the inverse matrix of [A])

Specifically, the whitening of the second moment of the output shows:

$$<\mathbf{V}(t)\mathbf{V}^T(t)>=[W][A]<\mathbf{s}(t)\,\mathbf{s}^T(t)>[A]^T[W]^T =[I] \quad (4)$$

This is equivalent to $[W] = [A]^{-1}$ provide that statistical de-correlation of sources

$$<\mathbf{s}(t)\,\mathbf{s}^T(t)> = [I] \qquad (5)$$

is true. If not, a whitening in the data domain is needed to get rid of the second order statistics, namely eliminating Gaussian random process and keeping high order information. In electrical engineering, this operation is known as the gain normalization of different sensor inputs. $\boldsymbol{U} = [Wz]\,\boldsymbol{x}$ where the zero-phase or symmetric matrix[BS95]

$$[Wz] = [Wz]^T = <\boldsymbol{x}\,\boldsymbol{x}^T>^{-1/2} \qquad (6)$$

can bc derived by setting

$<\boldsymbol{U}\,\boldsymbol{U}^T> = [Wz] <\boldsymbol{x}\,\boldsymbol{x}^T> [Wz]^T = [I]$,

and post-multiply with [Wz]: $[Wz]=[I][Wz]=[Wz] <\boldsymbol{x}\,\boldsymbol{x}^T> [Wz]^T[Wz]$, and canceling the common factor [Wz]: $<\boldsymbol{x}\,\boldsymbol{x}^T> [Wz]^T [Wz] = [I]$, where use is made of the symmetry to arrive at the result of inverse square-root of covariance matrix. Since [W] is the statistical inverse of [A], one can use it to obtain the unknown source point-by-point in the identical time order

$[W]\mathbf{X}(t)=[W][A]\,S(t) = S(t)$ (7)

While an ill-posed deterministic problem can not be uniquely solved, an ill-posed statistical problem has a lot more conditions in time to determine all those unknowns.

Deterministic: # of unknowns, S, A > # of known X
Stochastic computing the covariance R
# of unknowns = # known; data $R_{xx'}$ = source $R_{ss'}$
But if more than 2 sensors & de-correlated
$R_{ss'} = R_{ss}\,\delta_{ss'}$
then it's possible to gain more $Rx_1\,x_1$, $Rx_2\,x_2$, $Rx_1\,x_2$

This is not unlike the human experience, which by definition provides the statistics determination based on past experience. Real world applications are given in Part II, such as remote sensing subpixel composition, voice-dictation phoneme segmentation by means of ICA de-hyphenation, and cable TV bandwidth enhancement by simultaneously mixing all Sport and movie entertainment events.

The visual cortical feature detectors might be the end result of such a Redundancy Reduction Process (RRP), in which the activation of each feature detector is supported to be as statistically independent from the others as possible. Such as 'factorial code (of joint probability density)' potentially involves independence of all orders, but most studies have used only the second-order statistics required for de-correlating the outputs of a set of feature detectors. Field has observed that the early learning algorithms are mainly based on the second-order statistics, which might account for the missing opportunity. Current understanding is that the need of high order statistics such as the 4th order cumulant called Kurtosis may be captured completely by the information-Theoretical approach of maximum mutual information entropy underlying the Independent Component Analysis (ICA). The fourth cumulant, the Kurtosis *K(u)*, is often used by Helsinki's Oja group to seek the statistical matrix inversion.

$$K(\mathbf{V}) = <\mathbf{V}^4> - 3\,(<\mathbf{V}^2>)^2 \qquad (8)$$

because $<v_1v_2v_3v_4>_G = <v_1v_2>_G <v_3v_4>_G + <v_1v_3>_G <v_2v_4>_G + <v_2v_3>_G <v_1v_4>_G$ are reduced for identical process to (8). One considers a single weight vector update:

$$d\mathbf{w}/dt = dK/d\mathbf{w}. \qquad (9)$$

The other weight vectors are found by the projection pursuits. If each voice and image has its unique value of Kurtosis, then seeking a stationary Kurtosis yields the specific voice and image, without knowing what is the desired output: Gaussian, Laplacian, Multi-Modal Distribution & each has a Super-Gaussian, K>0, or Sub-Gaussian. K<0, Kurtosis value than Gaussian, K=0. Note that speech oscillation has Laplace distribution decaying exponentially from the mean value, zero amplitude, which is faster than Gaussian quadratic decay. Therefore, the Kurtosis of speech is called super-Gaussian and by definition, has a positive value (subtraction of smaller variance than that of Gaussian). On the contrary, an image histogram has a bimodal distribution for most grey scale images, then the variance is bigger than that of Gaussian. Therefore, an image has a negative Kurtosis value, so-called the sub-Gaussian.

Imagery edges occur naturally in human visual systems as a consequence of redundancy reduction towards "sparse & orthogonality feature maps," which have been recently derived from the maximum entropy information- theoretical first principle of artificial neural networks. Singularity edge-maps are sparse and orthogonal for the uniqueness & robust features necessary for pattern recognition tasks. Sparseness of singularity edge map needs more than second order statistics the ICA to extract it.

That decay of excitation patterns towards noisy outputs results in the stored memory eight matrix [W] among neurons. From the knowledge representation point of view, the more efficient and robust representation, the better. Two principles are the keys to achieve efficient representation: *orthogonality* and *sparseness* in the hits frequency of feature detectors leading to unique identification. For example, an edge-map with one's over zero background is clearly sparse, local, and almost orthogonal. The IC notion may be attributed first to Barrow in 1953 as the redundancy reduction process (RRP). In fact, the final IC State may be described by a factorized joint-probability density function, and is sometimes called as factorized code. This factor-code corresponds to all sparse orthogonal edge maps in the early vision processing. The second moment of IC must be by definition a diagonal matrix, which appears like a Gaussian random process whose off-diagonal elements cancel one another. If the learning of weight matrix [W] can achieves the maximum entropy *H(V)* of the output *V* or the linear slope portion of the maximum entropy sigmoidal neuron output $H(V) \cong H(\sigma(U)) \cong H(U)$ which implies that all nth moments of the ANN output components $U=\{u_1, u_2\}$ of two sensor neurons are independent in terms of the normalized statistical histograms $\rho(u)$ defined as: joint p.d.f.

$$\rho(x_1,x_2) = \rho(x_1)\rho(x_2)(1 + h(x_1,x_2))$$

Factorized Code [Ati92] implies:

$$E\{x_1{}^n,x_2{}^m\} = \int\int x_1{}^n,x_2{}^m \rho(x_1,x_2) \, d x_1 d x_2 = \int x_1{}^n \rho(x_1) dx_1 \int x_2{}^m \rho \, x_2) \, dx_2$$

$$= E\{x_1{}^n\} \, E\{x_2{}^m\}$$

Example: Gauss Center of Limiting Theorem

$$\rho(\xi) = \exp(-(\xi - <\xi>)^2/2\sigma);$$

$<\xi^4> = <\xi^2> <\xi^2> = 0$, if Gausissn iid

where

$$< u^n > = \int u^n \rho(u) du = E\{u^n\}$$

Biological evidence is first due to Nobel Laureats Hubel and Wiesel showing an oriented edge-map in the several octave scale in cats, similar to 2-D oriented Gabor Logon (information unit similar to a windowed FT or a WT without affine parameterization). Such an unsupervised learning methodology has been given in solving the statistically Blind Source Separation (BSS), as first introduced by C. Jutten, J. Herault.

$[w] = [[I] + [s]]^{-1}$ where $[s'] = [s] + \alpha f(y)g(x)$ is an odd function for Blind Source Separation (BSS) of Super-Gaussian Laplacian distribution of speeches $\Delta[W] = g(\mathbf{x})\tanh(\mathbf{u}^T)$ in terms of some ad hoc odd functions, in the first Snowbird ANN Conference in 1986. Both ICA subsequently coined by P. Comon [Com94], and BSS further elaborated by Herault & Jutten were appeared in Signal Processing Journal in 1991. Oja elaborated the nonlinear PCA learning, because neuron output $\mathbf{V} = \tanh(\mathbf{U}) = \mathbf{U} - 2/3\,\mathbf{U}^3 + \ldots$ has a similar Taylor expansion as $dK(\mathbf{V})/d\mathbf{w}$. The first principle of ICA may have several forms, e.g. absolute entropy versus mutual entropy, Neg-entropy-- the distance from the normality, Edgeworth versus Gram-Charlier expansions (of pdf in terms of moments) which are related to the maximum Shannon entropy H(**V**). The essential portion related to the change of weight matirx is equivalent in achieving the redundancy reduction toward independent components which gives rise naturally to a sparse orthogonal edge map (unfortunately only at one wavelet resolution). The landmark accomplishment of ICA is to obtain, by unsupervised learning algorithm, the edge-map as image feature $\vec{a}$, shown by Helsinki researchers using fourth order statistics of **V** -- Kurtosis K(**V**), and derived from information-theoretical first principle of ICA by Bell & Sejnowski. Amari has further contributed to the speedup of learning by suggesting a natural gradient descent, rather than the original entropy gradient involving a non-local weight matrix inversion.

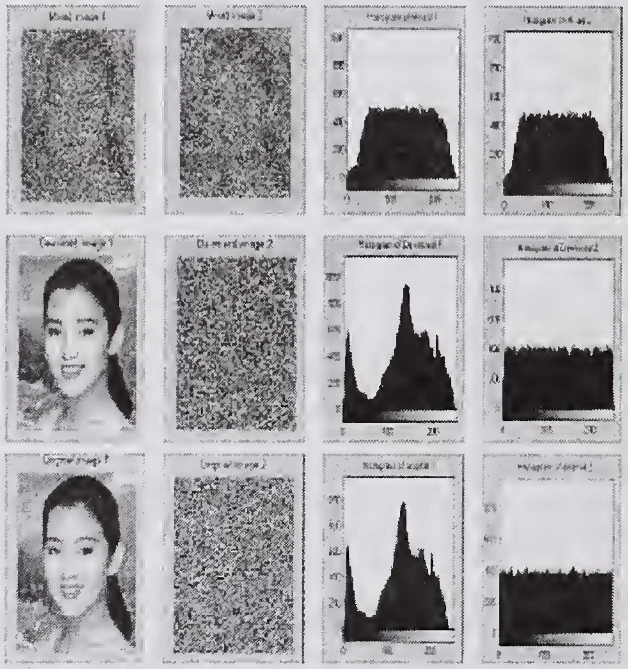


Fig. 4 Why do we have two eyes? They can provide instantaneous spatial learning without teacher, i.e. two eyes agree must be signal, and don't noise. A perfect denoise is possible using two eyes or two ears, two sides of tongue and two nose passages, two hands, etc., but one sensor needs the slower help from the brain memory itself. Sophisticate cross-talk de-mixing is in Sect. 5.

4. CYBERNATIC THEORY

Human intelligence can not yet be mathematically defined and addressed here. Instead, the supreme manifesto of the human intelligence might be the learning ability without teachers, which is modeled by the thermodynamics neural net learning theory. In so doing, we have discovered that the parallelism between the supervised associative recognition and the unsupervised ICA de-noise [1,2] (e.g. the cocktail party effect) is conveniently controlled by the Gibb's free energy temperature [3]. In this paper, we identify the temperature as the cybernetic temperature defined as a root-mean-square fluctuation of synaptic transmission activity.

During the last Russian Academy of Nonlinear Sciences Academician meeting (at St. Petersburg June 1999), I have proposed the role of cybernetic temperature for learning capability as warm blooded man, mammals, versus cold blooded dragons, reptile, lizards. Furthermore, I have investigated the information content of an unsupervised learning by means of Independent component analyses (ICA) with several students (cf. Wavelet Applications Orlando, April 2000 proceedings). This is based on joint probability density factorization for all independent moments (cf. Shannon's Principal Component Analyses (PCA) information content based on the second moment covariance only). Application to two eye de-noise, and image blind de-mixing and seven spectral band remote sensing are respectively given in [3,4,5].

Those who know of the animal intelligence having very little to do with brain sizes will be critical about the homeostasis theory having anything to do with the intelligence. Mathematically, neural network models of both learning seem to predict a constant temperature T for the minimization of the thermodynamic Helmholtz free energy, A = U - TS (equivalent ANN notation Lyaponov L = E –T H), in order to achieve the synergistic learning balanced between the supervised energy, $E(v_i)$, Hopfield-like minimization of neuron firing rate v_i, and the recently breakthrough of the unsupervised sensory pre-processing based on the output entropy $H(w_{ij})$ Bell-Sejnowski maximization. Since some mammals have bigger brains than Einstein's having a normal human being size, it implies not the brain size rather the interconnectivity wrinkles of the gray matter, which are responsible for associative memory. Again, it is not the degree of temperature rather the constancy of brain body temperature, which may be important to the kinetic diffusion rate controlling the chemical reactions that are vital to the healthy cellular functions (as evident in the excess fever causing by fatal diseases).

First, we define the supervised learning to include self-taught (involving higher motivation and intelligence) considered being equivalent to learning with a teacher either implicitly internally or explicitly externally. Secondly, we define the unsupervised learning to be the pre-attentive preprocessing of all real-time and short-term memory pairs of sensory inputs without conscience effort with associative recall. Neural network learning models suggest a constant brain cybernetic temperature, which balances the output energy $E(v_i)$ for neuron firing rates for supervised learning, and the maximum entropy $H(w_{ij})$ of synaptic matrix for unsupervised excitation decays for redundancy reduction (to wavelet or singularity-map). While the supervised learning (with implicit or explicit teaching) may be driven by the internal energy minimization, the unsupervised learning (sensory pre-processing) may be driven by the relaxation decaying processes by means of the maximization of local entropy. For example, there are 6.1 millions cones for color vision and 150 millions rods for dark light vision, and any imbalance on the visual neural pathway might cause the hallucination. This balance is achieved by the thermodynamics L = E – T H at a constant temperature T to determine the internal energy $E(v_i)$ minimization and the entropy $H(w_{ij})$ maximization. According to the theory of statistical mechanics, such a thermodynamic balance between E & H is possible due to a constant temperature T. This natural thermodynamic equilibrium might be useful to help develop fully the innate learning ability of a mammal. Thus, it is conjectured that the homeostasis of mammals must have a profound effect upon learning ability of the mammals, which in turn affect the development of intellectual capability. The cold blood reptiles can obviously learn without such a thermodynamic equilibrium, and it is interesting to notice a lower intelligence associated with them.

These models suggest it may also be important to the intercellular communication-mediated learning mechanism. Furthermore, based on recent breakthrough of sensory learning, the minimization of Helmholtz free energy L ≡E-TH at a constant T involves the internal energy E and the entropy H is believed to have maintained the thermodynamic equilibrium of those intercellular communication mechanisms useful for Hebbian synaptic modification. This free-energy minimization is mathematically shown to be Lyapunov functions that control a proper balance between the unsupervised sensory preprocessing based on maximum entropy of synaptic weights and the supervised learning based on minimization of neuron firing rate energy.

5. INFO-THEORETICAL MODEL

Recently, the biological edge map developed by the Nobel laureates, Hubel-Wiesel, was reproduced computationally by maximizing the neuron output entropy of among 10^4 images by means of maximum output entropy:

$$\partial H(\mathbf{V})/\partial[\mathbf{W}] = \partial[\mathbf{W}]/\partial t. \qquad (10)$$

Algorithmically, ANN adjusts [W] at the linear output range, $\mathbf{V}(t) = \tanh([\mathbf{W}]\mathbf{X}(t)) \approx [\mathbf{W}]\mathbf{X}(t)$, so that $<\mathbf{V}(t)\ \mathbf{V}(t)^T>_G = [I]$. Note that no decision of the sign of the tanh function is necessary in the linear range, implying that the maximal output entropy, and thus the input information is kept in [W]. There is a need to unify both the supervised learning of Principal Component Analyses (PCA) by Oja et al. and the unsupervised learning of ICA (advanced by Jutten & Herault, Comon (1991), Cardoso (1998) in France and by Bell-Sejnowski (1995) in U.S.A., and by Amari-Cichocki (1996) in Japan).

Baysian prob f(x,y) = f(x|y)f(y)= f(y|x)f(x)
Shannon Entropy H(x,y) = − < Ln f(x,y) >
=− <Ln f(x|y)f(y)> = H(x|y) + H(y)
=− <Ln f(y|x)f(x)> = H(y|x) + H(x)
= H(x) + H(y) − I(x,y)
Mutual Info I(x,y) = <Ln [f(x,y)/f(x)f(y)]>
=<Ln[f(x|y)f(y)/f(x)f(y)]>= −H(x|y)+ H(x)

Statistical information content similar to geometricalinformation content of PCA

The associative recall by the associative memory outer-product approach can determine the center of training set clustered around each ICA basis, (11), and only 30% of them are significant in the direction cosine sense, and the rest ICA bases have no significant alignment with the training set. This is similar to PCA eigenvalues, which fall off drastically after the principal components, and is called by Shannon as the degree of freedom of the information content. We have generalized information content to statistical information content for those non trivial ICA bases.

$$Define \quad [W] = \sum_i (\vec{w}_i \quad \vec{w}_i^{\,T}); \qquad (12)$$

A fast estimation of the principal information content of a normalized ICA basis is denoted similarly by the eigenvalue λ_k that sums the squared magnitude of all the projection of normalized training data X_i upon k basis :

$$\lambda_k = \frac{1}{M} \sum_{i=1}^{M} (\vec{x}_i, \vec{w}_k)^2 \qquad (13)$$

6. LYAPONOV CONVERGENCE PROOF

Szu has postulated the Helmhotz free energy [3]

$L(v_1,\ldots,v_n,w_1,\ldots w_n,)=E(v_1,,v_n)-T\,H(w_1,..w_n)$

as the Lyaponov function, and proves the convergence dL/dt < 0 of both supervised energy-E-minimization and unsupervised entropy-H-maximization dynamics in synergism: Given local gradients:

Min. energy: $du_i/dt_i = -\,\partial E/\partial v_i$

Max. entropy: $(\partial[w_i]/\partial t_i)=(\partial H/\partial[w_i])$.

Proof:

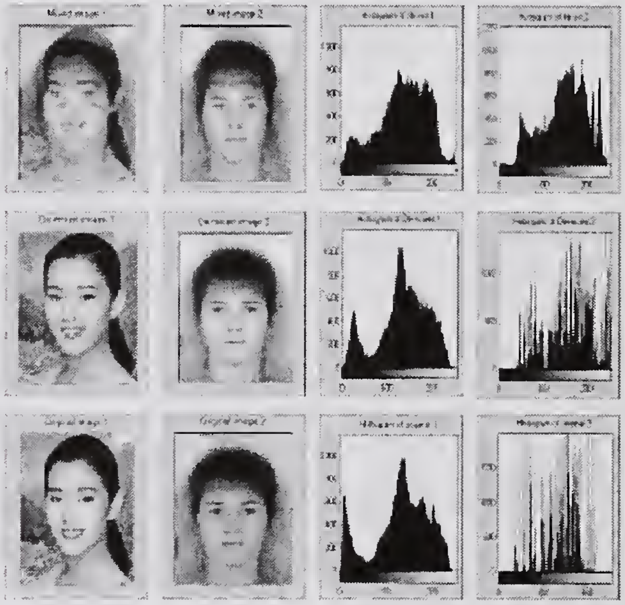
Min. Lyaponov (namely Helmhotz free energy):

$dL/dt = \Sigma_i (\partial E/\partial v_i)(\partial v_i/\partial u_i)\ (\partial u_i/\partial t_i)(dt_i/dt)-\ T\ \Sigma_i (\partial H/\partial w_i)(\partial w_i/\partial t_i)(dt_i/dt)$

$=-\{\Sigma_i(\partial E/\partial v_i)^2(\partial v_i/\partial u_i)+T\Sigma_i(\partial H/\partial w_i)^2\}(dt_i/dt)$

< 0 **Q.E.D.**

Here use is only made of stable cybernetic temperature T and the local gradient dynamic equations and positive firing rates to eliminate the temporal derivatives by spatial derivatives to form two real quadratic expressions, which by definition are always positive.



7 CONCLUSION

Helmhotz-Lyaponov drives the punctuated evolution for brain open systems at constant temperatures as opposed to less intelligent cold blood animals. One must go beyond the least mean square (LMS) error energy, and apply HOS to ANN. Applications are possible to multi-medium computers & machine intelligence.

8. REFERENCES

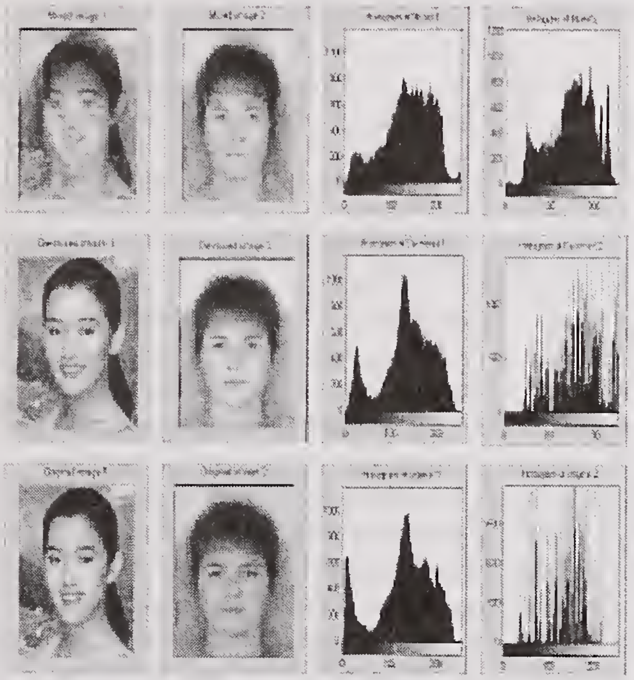
1 H. H. Szu, I. Kopriva, A. Peršin. Independent component analysis approach to resolve the multi-source limitation of the nutating rising-sun reticle based optical trackers, Optics Communications, Vol. 176, Issue 1-3, pp. 77-89, 2000

2 I.Kopriva, H.Szu, "Blind Discrimination of the Coherent Optical Sources by Using Reticle Based Optical Trackers is a Nonlinear ICA Problem", Proc. of the2nd Int. Workshop on Independnet Component Analysis and Blind Signal Separation,ed. P. Pajunen and J. Karhunen, June 19-22, 2000, Helsinki, Finland, pp. 51-56.

3 H. Szu, "Thermodynamics Energy for both supervised and unsupervised learning neural nets at a constant temperature," Int'l J. Neural Sys. Vol.9, pp. 175-186, June 1999.

4 H. Szu, "progresses in unsupervised artificial neural networks of blind image demixing, " New tech of IEEE Ind. Elec. Soc. Newsletter, pp. 7-12, June 1999.

5 H. Szu, "ICA-an enabling tech for Intelligent Sensory Processing", IEEE Circuits and Systems Newsletters December of 1999.



Properties of Learning Knowledge-Based Controllers

Edward Grant
Associate Professor
Director of the Center for Robotics and Intelligent Machines
Electrical and Computer Engineering Department
North Carolina State University
Raleigh, NC 27695

Gordon K. Lee
Professor
Assistant Dean for Research, College of Engineering
Mechanical and Aerospace Engineering Department
North Carolina State University
Raleigh, NC 27695

ABSTRACT

This paper addresses the field of knowledge-based systems, and in particular the sub-field of knowledge-based control systems. The rule-based approach used here, particularly in its machine learning or rule induction mode, continues as a major theme in the emerging field of data mining - the extraction of usable insights from large databases.

KEYWORDS: *knowledge-based control, learning control*

1. INTRODUCTION

The core tenet of this paper is that rule frameworks (i.e., directly programmed rule bases, rule bases derived by rule induction over experimental data and rule bases derived from induction over data produced by rule-based qualitative modeling with rule-based simulation) can be applied to achieve successful control of diverse systems.

The results obtained show that the underlying goals of the knowledge-based approach are as valid as ever and are particularly relevant to many of today's critical applications. In certain specific areas, they remain superior to all others. These areas include: (1) the inherent ability of knowledge-based systems to make their operation transparent to computer experts, to subject domain experts and to their users – a consequence of the systems' representing their knowledge directly in the form of symbolic rules; (2) the ability of the technology to capture the knowledge of the best experts in the field, to refine consistent and understandable symbolic rules from cases and empirical datasets; and (3) the ability to generalize such rules to cover a much larger set of possibilities than can feasibly be detailed explicitly by the domain expert or empirical dataset by using knowledge-based simulation, which is important in the field of diagnostic systems.

This, at its highest level, is what is demonstrated in this study. The authors methodology for designing knowledge-based (K-B) systems will be described, along with descriptions of its application to systems which can be to produce designs that proved successful in practice.

2. DEVELOPING KNOWLEDGE-BASED CONTROL

The focus of this paper is on mechanisms and technologies for implementing machine intelligence. Nevertheless the ability to learn must be one criterion for describing intelligent behavior. In robotics terms, intelligence is the ability of a machine to act autonomously in the presence of uncertainty. The ability of a robot to adjust its actions based on sensed information [1, 2, 3, 6, 12, 13, 14, 15] is another prerequisite for intelligence. In this work, the actions taken by the machine are considered to be intelligent if the actions reflect the action that a human would take, given the same conditions.

In advanced robotics systems [1, 2, 3, 6, 12, 13, 14, 15], robots are equipped with networked sensors: vision, tactile, proximity, speech recognition, voice synthesis, robot controllers, conveyors, vision processing equipment, and computers, an ideal domain for researching into machine intelligence (see Figure 1). However, the interconnection of physical systems, or the task undertaken by the system, does not make a machine intelligent. Intelligence comes from the manner in which the system is controlled or from the reasoning and decision making that the machine performs. In our terms, "intelligent control" is closely associated with "machine intelligence" [9, 10].

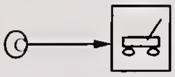
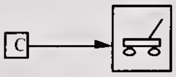
Schematic	Controller Type	MLC Location	Type of Controlled System
	Human	None	Simulator
	Auto	None	Simulator
	Human	None	Physical
	Auto	None	Physical

Figure 1. Automatic or Human Control (MLC - machine learned control)

Intelligent control systems must deal with sensor data and task specification and the task-state derived from integrated sensor data. An intelligent-control system must handle information about its own state and also the state of the environment; it must be capable of reasoning under uncertainty. Intelligent control commonly involves the use of both heuristic and algorithmic programming methods.

First we review hierarchically ordered control architectures for intelligent control. After this we concentrate on controlling dynamic systems with a variety of rule-based and machine learned programs. The final section deals with "human-in-the-loop" control as a knowledge-based controller.

3. ARCHITECTURES FOR INTELLIGENT CONTROL

Saridis [11] states that intelligent machines require the use of "generalized" control strategies to perform intelligent functions such as the simultaneous utilization of memory, learning or multi-level decision-making in response to "fuzzy" or qualitative commands. His work proposes that intelligent functions can be implemented using "intelligent control".

Intelligent control combines high-level decision-making, advanced mathematical modeling, and synthesis techniques of systems theory. These approaches along with linguistic methods attempt to deal with imprecise or incomplete information from which appropriate control actions evolve. The control functions in an intelligent machine have been implemented as a hierarchy of processes [1, 2, 3, 7, 11]. The upper layers concentrate on abstractions, decision-making and planning, while the lower levels concentrate on time-dependant sub-tasks, such as processing data from sensors or operating an actuator. Hierarchical decomposition is applied to complex control problems to reduce them to smaller sub-problems.

In the hierarchical control architectures of Albus [1, 2, 3] and Meystel [7], each layer essentially possesses the same processing nodes. These two architectures include a knowledge-base, sensory processing, task decomposition, and communication. But Saridis [11] recognized that each layer in a hierarchy need not perform the same activity over time and he and Albus [6] recognized that middle layers are frequently hierarchies of linguistic or heuristic decision structures that handle imprecise or "fuzzy" information. The National Institute of Standards and Technology (NIST) implemented Albus' architecture in manufacturing control (AMRF) [2] and in the NIST/DARPA Multiple Autonomous Undersea Vehicle (MAUV) [1]. Meystel [7] used an autonomous undersea vehicle as a demonstrator and Saridis [11] applied his architecture to space station robot and control applications.

4. REINFORCEMENT LEARNING

Since the mid-1970's, artificial intelligence (AI) methods have been continuously developed and applied by industry, business, and commerce. Expert systems are the most successful implementation of AI. However, the difficulties surrounding the development of the production rules for expert systems, going from the general to the specific led to the development of a sub-division of expert system technology known as "machine learning". In this section, we will look at the how the production rules for "rule-based" control can be produced manually and automatically and we will discuss approaches for achieving machine -learned control (MLC). We will describe a controller based on the machine learning algorithm BOXES [10], an algorithm that uses a reinforcement learning approach and we will discuss the implementation of neural networks for control. Both reinforcement learning and competitive learning are considered [13].

$$\ddot{\theta} = \frac{g\sin\theta - \cos\theta\left[\dfrac{F + m_p l\,\dot{\theta}^2\sin\theta}{m_c + m_p}\right]}{l\left[\dfrac{4}{3} - \dfrac{m_p\cos\theta^2}{m_c + m_p}\right]}$$

$$\ddot{x} = \frac{F + m_p l\,[\dot{\theta}^2\sin\theta - \ddot{\theta}\cos\theta]}{m_c + m_p}$$

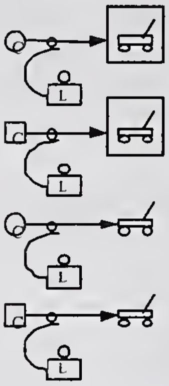
Figure 2. The Pole and Cart Equations

Rule-Based Control

Humans are capable of deriving a set of control rules through the process of interpretation. For example, consider the equations for the pole and cart problem (see Figure 2). Makarovic [8] derived a rule by examining the system's differential equations of motion (see Figure 3). The Makarovic rule worked well when the parameters of the system remained constant. When system parameters changed, the Makarovic rule cannot guarantee success. This showed that the arbitrary choice of one set of threshold values is not ideal for a system whose configuration changes. In contrast, a rule derived from observing a physical system's performance [18] can be written without any threshold values placed on the observation. Here the condition part of the rules only deals with the sign of the errors and with the sign of the variations of observed system state variables. This approach reflects human control heuristics and it can adapt to varying system configurations.

```
if theta_dot > THRESHOLD theta_dot then push RIGHT

if theta_dot > -THRESHOLD theta_dot then push LEFT

if theta > THRESHOLD theta then push RIGHT

if theta < -THRESHOLD theta then push LEFT

if x_dot > THRESHOLD x_dot then push RIGHT

if x_dot < -THRESHOLD x_dot then push LEFT

if x > THRESHOLD x then push RIGHT

if x < -THRESHOLD x then push LEFT
```

Figure 3. The Makarovic Rule Derived from Interpreting the Equations of Motion



Schematic	Controller Tvpe	MLC Location	Tvpe of Control System
	Human	Off-line	Simulator
	Auto	Off-line	Simulator
	Human	Off-line	Physical
	Auto	Off-line	Physical

Figure 4. Machine-Learned Control (Passive Learning)

Machine Learned Control

Machine learning as presented here is classified into two areas: (1) artificial-intelligence type learning based on symbolic computation and (2) neural nets. These are chosen because we have first-hand experience of applying them to real-world applications. An effective machine learning system must use sampled data to generate internal updates and also be capable of explaining its findings in an understandable way, e.g., symbolically. The learning system must also be

```
if (theta(k) > THRESHOLD)
 then
 if ((theta(k) < theta(k-1))
    and (|theta(k) - theta(k-1)| > |theta(k-1) - theta(k-2)|))
  then
    apply a RIGHT force
  else
    apply a LEFT force

if (theta(k) < -THRESHOLD)
 then
 if ((theta(k) > theta(k-1)
    and (|theta(k) - theta(k-1) > |theta(k-1) - theta(k-2)|))
 then
   apply a RIGHT force
 else
   apply a LEFT force

if (|theta(k)| <= THRESHOLD)
 then
 if(x(k) >= 0)
  then
    if ((x(k) < x(k-1)) and (|x(k) - x(k-1) - x(k-2)|))
    then
      apply a LEFT force
    else
      apply a RIGHT force
if (x(k) < 0)
 then
  if ((x(k) > x(k-1)) and (|x(k) - x(k-1)| > |x(k-1) - x(k-2)|))
    then
     apply a RIGHT force
    else
     apply a LEFT force
```

Figure 5. A Control Rule Derived from Experimentation with a Pole and Cart Simulator

able to provide an explanation of its results to a human-expert. The findings should also improve the human expert's understanding and verification. Artificial-intelligence type learning originated from an investigation into the possibility of using decision trees or production rules for concept representation. Since then, the work has extended to the use of decision trees and production rules to handle most conventional data types, including noisy data sets, and as a knowledge acquisition tool (see Figures 4 and 5).

Reinforcement Learning

Reinforcement learning, of the type produced by Michalski [9] and Michie [10], is similar to feedback for adaptation. However, unlike supervised learning, reinforcement feedback learning only gives an indication of the value of the system's action. Reinforcement is a feedback on the correctness of an action; it is not information on what the correct action is. Reinforcement learning is useful in cases where supervisory information is not available (see Figure 6).

Also, reinforcement learning falls into two categories: (1) non-associative type, which only receives a reinforcement signal from the environment and (2) associative reinforcement learning, where the system receives both a reinforcement signal, and sensory information, on the state of the environment. Sensors are used to discriminate between different situations. This we considered more suited to our particular needs with the pole and cart application. We will discuss the application of rule-based (MLC) and neural network controllers to control a pole-cart system by using AI techniques.

```
theta_dot > 0 : RIGHT
theta_dot <= 0 :
    theta <= -2 : LEFT
    theta > -2:
        thata_dot <= -1 : LEFT
        theta_dot > -1:
            x_dot <= -6:
                theta <= 1: LEFT
                theta > 1 : RIGHT
            x_dot > -6 :
                x <= 0 : LEFT
                x > 0 : RIGHT
```

Figure 6. A Control Rule for the Pole and Cart-Derived Using the BOXES Algorithm

'BOXES'

In Michie and Chamber's learning algorithm 'BOXES' [10, 13], the physical state space is partitioned into boxes. The algorithm learns to set correct decisions for each box through trial-and-error [10, 13]. Unfortunately, state space partitioning prior to experimentation is arbitrary because it is reliant on human knowledge. If the original partitioning is wrong, the algorithm can not learn to correct it. In the following, we will show how our rule can be used to partition the state space in the pole-cart application.

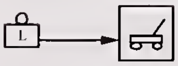
Schematic	Controller Type	MLC Location	Type of Controlled System
	MLC	On-line	Simulator
	MLC	On-line	Physical

Figure 7. Machine-Learned Control

5. NEURAL NETWORK - BASED REINFORCEMENT LEARNING

A learning controller consisting of a two-layered neural network was used to implement the input-output transfer function and an evaluation network, a look-up table, which provides the necessary reinforcement signal for evaluative feedback via a goal oriented performance index. The high-level architecture for the teaching controller is given in Figures 7 and 8.

Neural Networks

Neural networks implement information storage with synaptic weights storing information and distributed patterns acting as keys; they combine the benefits of both the computational method and look-up tables. With neural networks,

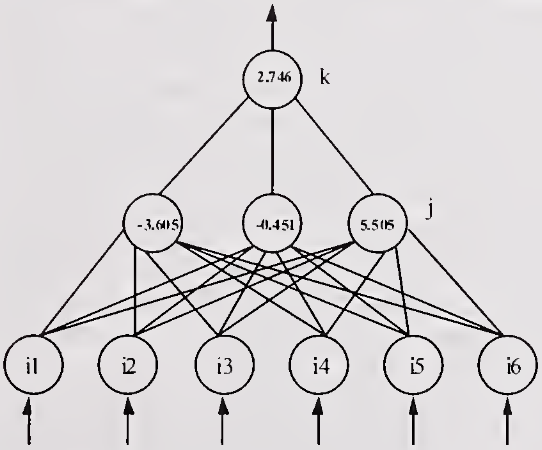


Figure 8. A Neural Network Controller for Controlling a Pole and Cart

Pattern Number	Pattern Features u1 u2 u3 u4 u5 u6	Actual Output	Classification
1	0 0 0 0 0 0	0.006910	0
2	0 0 0 0 0 1	0.003990	0
3	0 0 0 0 1 0	0.002360	0
4	0 0 0 0 1 1	0.001337	0
•	•	•	•
•	•	•	•
32	0 1 1 1 1 1	0.063977	0
33	1 0 0 0 0 0	0.959916	1
34	1 0 0 0 0 1	0.958932	1
•	•	•	•
•	•	•	•
64	1 1 1 1 1 1	0.952255	1
65	0.5 0 0 0 0 0	0.819989	1
66	0.5 0 0 0 0 1	0.616265	1
•	•	•	•
•	•	•	•
96	0.5 1 1 1 1 1	0.824466	1

Figure 9. A Rule-Based Table Look Up for Determining Neural Network Control Actions

information about control situations is coded in terms of distributed patterns; hence they can support distributed representation and reduce the storage requirements associated with control surface dimension.

A neural network can specify control actions for a given situation not visited during learning; it specifies according to its similarity. This associated structure automatically generalizes according to degree of similarity. The trade-off between computation time and storage space is resolved using neural networks.

Establishing the Look-up Table

In reinforcement learning, at every time step during learning, control actions are evaluated with respect to a sub-goal. The action that maximizes the sub-goal is regarded as the optimal control action and is rewarded; all other actions are punished. In the pole-and-cart, learning is difficult because the effects on choosing different actions cannot be tested. So, here we evaluate alternative control actions with respect to a small region of the state space. We also assume that they have the same reinforcement value (see Figure 9).

Decoding the State Variables

The term "decoder" describes the process of accepting an input situation and transforming it into one activity from a choice of a large number. Hence evaluation signals are stored as a look-up table where an input situation appears as an activity on a single path-way to a storage location. The storage location contains the appropriate evaluation specification. This approach was motivated by 'BOXES' [10]. Here, the four-dimensional state-space is divided into disjoint regions ('BOXES') by quantizing the state variables. The evaluation of different control actions was made with respect to a sub-goal. This estimate provided the necessary reinforcement signal for a reinforcement learning neural-network (RLNN) for control [13].

6. LOUGHBOROUGH GLUE DISPENSING WORKCELL

After the researchers at Loughborough University of Technology had tried numerous methods to visually inspect a dispensed 'blob' (e.g., inspecting for the "blob volume" using striped light) it was found that a simple feedback measure gave acceptable control. Tests using the "blob area" as the measured variable showed that this parameter could keep the process within a desired operating bandwidth. The measured variable based on the blob area, termed "box-area-ratio" (BAR), could distinguish between many of the common faults associated with the process.

Common faults observed in the process include: (1) a blob collapsing; (2) stringy (stitching) attachments; and (3) the presence of entrapped air. The researchers used on-line data recorded from the process to elicit information from that data. The extracted information is then presented as rules. After the process expert had verified these rules, the rules became a "knowledge-based" controller. During this period of interaction with the process expert, an interesting observation was made; it appeared that the expert measured the performance of the process through fault diagnosis (Williams, West,

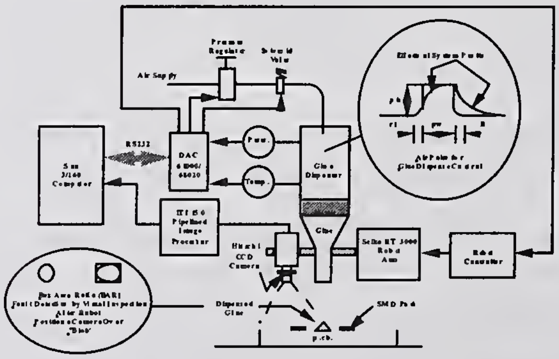


Figure 10. A Schematic of the LUT Glue Dispensing Workcell

and Hinde (1992) [12, 13]). This observation led to the development of a knowledge-based controller, one that was tested on the process (see Figure 10).

Before a knowledge-based controller can produce the rules inherent or embedded in the process data, a model must be established. Such a model produces rules from process data (Shepherd (1992) [12]). The model was constructed in three parts: image capture, feature extraction, and classification (see Figure 11).

The data from the LUT process was first normalized, so that it matched the integer requirement of the rule induction software. Thus, a blob area of 1103 is equivalent to a normalized area of 1.103. Also, since the bubble threshold is set for a 10% increase for the 'blob area', the data are classified as bub_inc. When there is more than one fault present in the data, each encountered fault is recorded as an attribute exceeding a threshold. Multiple fault conditions are obtained when a combination of attributes has exceeded their limits and an attribute-class vector is repeated for every exceeded threshold. Every exceeded threshold is represented as new class vector and they are added to the class list. Note that in order to be consistent, these may also combine any original class.

```
IF BAR > threshold THEN
  IF area_diff < bubble_threshold THEN
    IF area outside control limits THEN
      apply rule-based control action
    ELSE
     do nothing
IF risetime > risetime fault threshold THEN
  flag air
    IF falltime > falltime fault threshold THEN
      flag pulse width and pulse height
```

Figure 11. Pseudo-code of Process Operators Control Rule

Two knowledge-based controllers were tested on the LUT adhesive dispensing process: (1) a controller based on operator derived rules alone, and (2) a controller based on the VACLS rule induction algorithm (control via the fault detection rules derived from the process data). A simple BANG_BANG controller was written in C; this was used to maintain the dispensed blob area within 5% of a target area of 30,000 pixels. The results were remarkable in that the knowledge-based controller using the VACLS rule induction algorithm was highly successful in dealing with this application environment that is inherently difficult to control, and where knowledge elucidated from the expert human operator proved crucial, when combined with rule induction over empirical data from the multiple sensors. These results reinforced the value of human-in-the-loop type controllers whereby information from the human along with results from simple experimentation improves system performance.

6. CONCLUSIONS

Knowledge-based controllers (e.g., those constructed using expert verified rules) were tested on the various dynamic systems including the LUT industrial process control system. The control experiments tested overall system performance based on data from dynamic system and process parameters. To extract the knowledge-based control rules experiments were conducted on simulators and physical systems. Human control with simulators is achievable, but difficult with physical plant with fast response times. Passive learning proved useful but machine learned control had limitations, particularly when used with physical systems. In terms of the LUT process this included the variation of area and measured and programmed pulse height variation.

Two factors that have to be taken into account when using rule induction algorithms are timing, e.g., what are the overheads associated with implementing rules, and clashes. These influence of these two factors is reduced if the following procedure is adopted when preparing data prior to submitting it to the algorithm being used. First, divide the data set into two and train the algorithm on one half of the data set. Second, build the rule-based controller and test its performance on the other half of the data set. Third, after refining the controller, install it into the process and obtain test results. This was the procedure adopted when working with VALCS and it produced the richest amount of information.

Although this amount of information may not be wholly necessary for controlling a process, it does aid the expert in understanding the process performance and focuses on important inter-relationships. The importance of clashes is that they present a logical interpretation for fault monitoring and diagnosis. Clashes aid the expert understanding of the flags that are set as the process operates.

ACKNOWLEDGEMENTS

The authors would like to thank the faculty and students of the College of Engineering at North

Carolina State University, the Department of Computer Science at Strathclyde University, the Turing Institute, British Aerospace (HOTOL). Also, thanks go to Drs. David Williams and Andrew West of the Dept. of Manufacturing Eng., at Loughborough University of Technology.

7. REFERENCES

[1] Albus, J. S., "System Description and Design Architecture for Multiple Autonomous Undersea Vehicles", NIST Report 1251, Gaithersburg, MD, September 1988.

[2] Albus, J. S., McCain, H. G., and Lumia, R., "NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)", NIST Technical Note 1235, 1989.

[3] Albus, J. S., “A Theory of Intelligent Systems”, in Proceedings of the 5th IEEE International Symposium on Intelligent Control, Eds: A. Meystel, J. Hereth and S. Gray, 5-7 September, Philadelphia, PA, USA, pp 866-875, 1990.

[4] Efstathiou, J., Davies, B., Razban, A., and Harris, S., “Expert Systems for an Adhesive Dispensing Robot”, in Proceedings of the Sixth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE’93), Edinburgh, Scotland, pp 94-97, 1993.

[5] Grant, E., “Machine Learned Control”, Final Report to the NEL/BAe/BT/SERC, The Turing Institute, Glasgow Scotland, UK, 1992.

[6] Grant, E., *The Knowledge-based Control of Robot Workcells and Dynamic Systems*, Ph.D. Dissertation, University of Strathclyde, 1999.

[7] Isik, C. and Meystel, A., "Pilot Level of a Hierarchical Controller for an Unmanned Mobile Robot", *IEEE Journal of Robotics and Automation*, Vol. 4, June 1988, pp 241-255, 1988.

[8] Makarovic, A., “A Qualitative Way of Solving the Pole-balancing Problem”, *Machine Intelligence 12*, J. E. Hayes, D. Michie and E.Tygu (eds.), Oxford University Press, Oxford, England, 1989.

[9] Michalski, R. S. and Larson, J., “Incremental Generation of the VL1 Hypotheses: the Underlying Methodology and the Description of Program AQ11”, Technical Report: ISG 83-3, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, ILL, USA, 1969.

[10] Michie, D. and Chambers, R. A., “BOXES: An Experiment in Adaptive Control”, *Machine Intelligence 2*, E. Dale and D. Michie (eds.), Oliver and Boyd, Edinburgh, Scotland, 1968.

[11] Saridis, G. N., "On the Revised Theory of Intelligent Machines", CIRSSE Report 58, ECSE Department, Rensselaer Polytechnic Institute, Troy, NY, 12180-3590, USA, 1990.

[12] Shepherd, B. A., *High-level Programming of Vision Guided Assembly Tasks*, Ph.D. Dissertation, University of Strathclyde, 1992.

[13] West, A. A., Williams, D. J., and Hinde, C. J., “Experiences of the Application of Intelligent Control Paradigms to Real Manufacturing Processes”, in Proc Instn Mech Engrs, Vol. 209, pp 293-308, 1995.

[14] Williams, D. J. West, A. A. and Hinde, C. J., "A Discrete Process Control Software Testbed", Science and Engineering Research Council Report, Grant GR/F 37101, 1992.

[15] Williams, D. J. West, A. A. and Hinde, C. J., "The Selection of Software Techniques for Discrete Process Control Applications", Science and Engineering Research Council Report, Grant GR/F 71973 1992.

[16] Zhang, B., *Experiments in Learning Control Using Neural Networks*, Ph.D. Dissertation, University of Strathclyde, 1991.

Assessing the Run-Time Performance of Artificial Intelligence Architectures

S. A. Wallace, J. E. Laird, and K. J. Coulter
University of Michigan
1101 Beal Ave.
Ann Arbor, MI 48109-2110

ABSTRACT

As intelligent systems are pushed forward to become more autonomous, there is a tendency for the underlying software architecture to grow in complexity to support these new behaviors. However, with the addition of new features, two potential costs may be incurred: increased execution time and additional memory requirements. As architectures evolve, it is important to continually evaluate the costs and benefits of each new change. Seemingly very similar architectures may require significantly different resources; small changes to the features in a single architecture may have a large impact on its performance. Thus, it is necessary to understand and to quantify the resources consumed by different architectures and by the components of a single architecture. Unfortunately, there is no standard method for evaluating features of an architecture or for comparing sets of architectures. In this paper, we begin by discussing such a methodology. We then dissect the Soar architecture into a core set of functionality and examine how incrementally adding each of the features found in the original implementation affects the overall performance and resource requirements. Next, we show how the same methodology can be used to compare two different architectures. Finally, we discuss initial results of a comparison that indicates both qualitative and quantitative differences between the Soar and CLIPS architectures.

KEYWORDS: *Architecture evaluation, Soar, CLIPS*

1. Introduction

As artificial intelligent agents become increasingly robust and autonomous, the software underlying their behavior also becomes more and more complex. Success with simple agents in simple domains inspires research into the capabilities required to operate more efficiently and effectively. This in turn causes the software architectures to evolve, as functionality is added to support the new demands. Because this is a common process, many architectures have been developed incrementally over the course of many years as they become increasingly sophisticated.

Design decisions made at implementation time often play critical roles in the efficiency (both in time and space complexity) of the architecture. The impact is seen both when the new features are used by an agent and in some cases even when the features are not used. Thus, after a feature is added to an architecture, agents operating in complex domains and relying heavily on the new feature may operate more efficiently than was previously possible, while agents that do not rely on the new architectural feature may become less efficient. To properly assess the impact of an architectural modification, it is necessary to quantify the resource consumption of that modification. In most cases, it is extremely difficult to draw meaningful conclusions using analytical methods, although in some cases, a formula that relies on prior knowledge of a relatively few variables may be obtainable. Even in these instances, however, comparisons between two such formulas are further hampered by the fact that constant factor differences may have profound implications on their relative suitability in real-world tasks. As a result, we believe that empirical methods are currently the most suitable way to evaluate the impact of design decisions.

Additionally, two distinct agent architectures are likely to yield agents with differing efficiencies even if the architectures (and agents) appear otherwise similar. As architectures become increasingly divergent, it may become overtly obvious that the features of one architecture are better suited to a particular task than are those of another. In many cases, however, this is not necessarily clear a-priori. As a result, designers of intelligent agents, and of agent architectures may benefit from understanding the relative differences in resource consumption between two or more architectures. As in the single architecture case, empirical methods can yield approximate answers to these questions. Unfortunately, however, there are no standard methodologies for evaluating the resource consumption of a particular architecture or of components of a single architecture.

In this paper, we discuss a methodology that can be used to examine the resource requirements of an architecture as a whole, or of particular aspects of that architecture. We present a practical example by applying this methodology first to components of the Soar architecture and then to the standard version of both the Soar and CLIPS architectures. Our results show both qualitative and quantitative differences between these two architectures and show how components of the Soar architecture contribute to its overall performance. Early versions of some of the material and results in this paper appeared in [15,16].

2. Architectures, Knowledge and Modularity

The class of AI symbolic architectures we are interested in are those that support the development of general, intelligent knowledge—rich agents. Following Newell's description [10], an architecture is the fixed set of memories and processing units that realize a symbol processing system. A symbol system supports the acquisition, representation, storage, and manipulation of symbolic structures. An architecture is analogous to the hardware of a standard computer, while the symbols (which encode knowledge) correspond to software. The role of a general symbolic architecture is to support the representation and deployment of diverse types of knowledge that are applicable to various goals and actions.

The basic functions performed by an architecture usually consist of the following (from Newell [10] p. 83):

- The fetch-execute cycle
  - Assemble the operator and operands
  - Apply the operator to the operands using architectural primitives
  - Store the results for later use
- Input and output

Architectures are distinguished by their implementation of these functions, and the specific set of primitive operations supported. For example, many architectures choose the next operator and operand by organizing their knowledge as sequences of operators and operands, incrementing a program counter to select the next operator. They also have additional control constructs such as conditionals and loops, but depend on the designer to organize the knowledge so that it is executed in the correct order. Other architectures, such as rule-based systems, examine small units of knowledge in parallel, selecting an operator and operands based on properties of the current situation. Some examples of these architectures inlude: Atlantis [4], CLIPS [1], Soar [7] and PRS [6].

Because the definition above leaves a fair amount of room for interpretation, architectures can often be further distinguished by the inclusion of additional functions, such as interruption mechanisms, error-handling methods, goal mechanism, etc. The inclusion of such functionality illustrates the necessarily blurry distinction between knowledge and architecture. Because most agent architectures are Turing complete, features not supplied directly by the architecture can often be emulated by the appropriate addition of knowledge, but with additional execution time overhead. However, it is often unclear a-priori how different design decision will affect future performance, and designers may choose to construct architectures modularly.

Architectures are modular in so far as features can be removed while still preserving the basic requirements of an architecture. Potentially, modules can be added or removed in order to optimize the architecture for a particular situation. Note that this is different from simply being able to refrain from using certain features because it suggests that the internal design of the architecture with and without a modular feature is different.

3. A Methodology for Agent Architecture Evaluation

Our methodology begins with dissecting the architecture into constituent modules, leaving a core set of features intact. In many cases, such as when an architecture is developed incrementally, certain features may be naturally modular. In other cases, a great deal of thought may be required to determine what aspects of the architecture can be removed while still allowing the core functionality to meet the design goals of the researchers. In either situation, modifications to the source code will undoubtedly be necessary to construct a set of architectural variants that combine different modular features with the core functionality.

The second step in our methodology consists of determining a class of situations in which to examine the architectural variants. Particularly interesting problem classes may be found at both ends of a spectrum from situations that do not rely on a specific architectural feature to those which rely very heavily on such a feature. Although any single study is likely to be limited to examining a relatively small problem class, as the number of studies increases, it is anticipated that general trends will emerge indicating which architectural variant is most suited for a particular class of problems.

The third step involves selecting an environment in which to examine the problem class selected in the previous step. Because there is no single environment that can be used to represent "environments" as a whole, selection must be made with care, and equal care must be used to ensure that results are not over generalized. Understanding how the environment fits within a typical taxonomy (e.g. from Russell and Norvig [12]) may help moderate this problem.

Fourth, for each architectural variant, an agent must be designed to solve the specific problem within the selected environment. Agents solving the same problem form a group. All agents within a group must utilize the same problem solving methods. The effect of this constraint is that any two agents within a group must not only have identical interactions with the environment, but must also utilize the same internal problem-solving methodology. Proper implementation of this step is critical; otherwise there is a serious risk of confusing the contribution of different architectural aspects and different knowledge (i.e. problem solving methods) on the overall results. However, in certain circumstances this pitfall is eliminated because all of the agents within a group can be implemented using identical knowledge. This exceptional case occurs when architectural variants differ only in their inclusion or exclusion of unused features. Once a group of agents has been fully implemented, the performance of agent/architecture pairs can be directly compared.

4. Soar and Its Modular Components

The Soar [7] architecture is a forward chaining production system based on the RETE matching algorithm [2,3]. It contains a long-term memory (LTM) that stores production rules, and a short-term memory (STM) containing elements that are matched by the rules.

Short term, potentially volatile, knowledge is stored in STM in the form of a directed graph with labeled edges. Each memory element can be thought of as an ordered triplet whose slots refer to the parent node, the edge name and the child node respectively. Because this structure is so generic, it can be used to represent a multitude of more complex data structures.

Long term, stable knowledge, is stored in LTM as a set of productions. Productions are created explicitly by the programmer, or may be generated automatically by Soar's learning mechanism. The condition of a rule may contain either variables or constants, and variables may be bound to any of the three slots in a memory element's ordered triplet. This ability allows a large amount of flexibility in terms of how a rule is designed, but it can also greatly increase matching costs when it is used indiscriminately. The condition side of Soar's rules may also ensure that values bound to a variable satisfy one or more basic predicates (e.g. >, <, =). Generic predicates, however, are not supported in a rule's conditions. The right hand, or action side of a rule, can be used to modify the contents of STM. Additionally, it can propose architectural-constructs called operators or preferences for such operators.

In Soar, knowledge is deployed by rule firings. This process begins as follows:

- First, determine which rules, if any, match the current contents of STM.
- Next, fire all matching rules in parallel, by executing the instructions in their right hand side.

These two steps, called an elaboration cycle, are repeated until a quiescent state is reached in which no more rules can fire. Parallel rule firings allow Soar to make use of all relevant knowledge in a given circumstance. It also forces programmers to explicitly encode control knowledge into rules to select operators instead of relying on a potentially cryptic architectural mechanism to determine which rule among the current matches should actually be fired.

In addition to the basic execution supported by the elaboration phase, Soar also has an architecturally supported decision-making phase that occurs immediately after elaborations have ceased. During the decision phase, operators representing actions of higher-level goals, which have been proposed during the elaboration phase, are examined. The operators are ranked according to their relative preferences, which have also been specified during the elaborations. At this point, Soar selects the operator with the highest preference to be pursued. Although the serial nature of pursuing operators may seem similar to productions systems that fire rules serially, this is not typically true. One important distinction is that in Soar, knowledge about proposed operators is explicitly declared, and is available to be used for further reasoning, whereas information about matched rules in a serial system is typically not available to be used in this way.

In certain cases, Soar may decide that it is no longer making progress on the current problem (e.g. the elaboration phase terminates without any rules being fired, or Soar cannot select between two operators). In such cases, Soar will react by creating a sub-state in which further reasoning can take place. Within this sub-state, operators can be proposed and pursued just as in the super-state. The sub-state vanishes when Soar has done enough reasoning to resolve the problem that triggered its creation. It is during the resolution of sub-states that Soar's learning mechanism creates new search control knowledge (in the form of a rule) and adds it to LTM so that similar sub-states (and the additional reasoning to resolve them) can be avoided in the future.

Although Soar has been developed incrementally over a number of years, the mechanisms needed to modularize the architecture were not completely in place. Nonetheless, some features were clear candidates for modularization, and these are listed below:

- Detailed Timing Facilities - Soar has the ability to keep track of the time spent on various aspects of execution, but in many cases this information is not critical to the task.
- Callbacks – Soar has the ability to call user-defined functions during execution. Some of these callbacks are invoked many times per decision cycle, and even if no functions are registered with the architecture, some overhead is incurred due to looking up and testing one or more variables.
- Learning - Each time Soar completes reasoning within a sub-state, the architecture has the ability to learn a new rule. When using Soar in certain domains, however, learning has not been employed because these forces have been expected to perform at an expert level without undergoing a potentially costly training phase.
- Backtracing Mechanism - Soar also has the ability to keep (potentially elaborate) information as to how it reached a particular conclusion. The full power of this feature is used only during learning. Thus, as only a small portion of this mechanism is required for other purposes, significant amounts of source code can be removed or optimized when learning is also removed.

The four features we have identified above are only a subset of the features in the Soar architecture that could be modularized. However, this partitioning of the architecture was particularly suitable for our initial exploration because in certain testbed environments, a single set of knowledge could be used to examine all of the resulting architectural variants.

Three variants of the Soar architecture were examined for our tests by including or excluding some or all of the modular features described above. Variant 1, which we will also refer to as the standard version of Soar, includes all of the modular features. Variant 2, removes the Detailed Timing Facilities as well as the Callback module. Variant 3 removes all of the modular features described above.

5. Decision-Making Strategies

The class of problems we have selected for the initial implementation of our methodology is what we refer to as decision-making strategies. Most, if not all, agents are similar in that they must examine their current state and decide which of the many possible options to pursue. This process can take place in a variety of ways. In particular, one set of methodologies that can be used by Soar (as well as by a potentially large set of agent architectures) focuses on the individual pieces of knowledge which must be brought to bear in order to make the most appropriate decision about the next action. Some agents, for example, may use knowledge that directly ties a particular state or set of states to the most appropriate action. If the preconditions for each action are disjoint, only a single piece of knowledge will be brought to bear in any given situation, and the decision will essentially make itself. This is analogous to the operation of a lookup-table. Other agents may bring multiple pieces of knowledge to bear in order to make their decision. As the knowledge becomes hierarchically organized, the agent will go through an increasing number of refinement steps (reflected by a path in the tree from the root to a leaf) before it is able to select the most appropriate action for the circumstances. It is this general process of refinement that we have used as the basis for this study. Below is a list of decision-making strategies in which the refinement process is increasingly complex:

- Simple, Declared Actions - Actions are represented declaratively to the system, in Soar this is done using operators. The programmer supplies enough knowledge to guarantee that only one action is applicable at any given time, thus no conflicts between courses of action can arise.
- Three-Phase Decision - The decision takes place in three distinct phases. In the first of these, actions are proposed, in the second phase actions are ranked according to their relative preferences and finally the most preferred action is selected and pursued. This allows for multiple layers of refinement in the decision making process, potentially decreasing the size and complexity of the knowledge base.
- Goal Directed - A goal is a subtask that requires the application and pursuit of a sequence of one or more actions. In this strategy, goals are selected the same manner as primitive actions and may improve performance by constraining the subsequent problem solving. Soar expresses goals with high-level operators, and uses sub-states to perform the reasoning needed to achieve these goals.

6. Towers of Hanoi

The Towers of Hanoi problem is well known to the AI community and has an equally well-known optimal solution. Although it is a relatively simple problem, it is complex enough to examine the class of decision-making strategies outlined in the previous section. Moreover, within this domain it is possible to limit differences between the agents' knowledge to exactly what is required to implement each decision making strategy. It is important to remember that we intend this environment to be used as a starting point for further investigation, and as a proof of concept. No single domain can claim to be representative of all situations an agent may face in general.

Table 1 shows the runtime performance of the Soar architectural variants described in section 4. Across all problem-solving strategies, significant timesavings are achieved between variants 1 and 2 as unused features are removed from the architecture. Further savings are achieved in the Tower of Hanoi subgoaling agent because the differences between variants 2 and 3 affect the efficiency of the architectural subgoaling process in situations where learning is not employed. Based on these results, and knowledge of how the architecture was modified, we expect that all Soar agents that do not require learning will achieve some performance savings by using the more streamlined architectural variants. Moreover, we further expect that agents that solve problems similarly to the Towers of Hanoi subgoaling agent above will be most enhanced. That is, agents that use a large number of subgoals, each of which requires relatively little reasoning to resolve on its own.

Variant	Declarative	3-Phase	Goal-Directed
Standard Soar	12.45	21.98	22.17
Variant 2	4.19	11.06	7.92
Soar LITE	4.13	11.42	5.64

Table 1. Soar Performance in Towers of Hanoi

7. Complex Real-Time Task: Quake II

The tests we conducted in Section 6 seemed to indicate that a substantial savings could be gained in situations that do not require learning. To substantiate this belief, we looked for other previously developed agents that shared this attribute and could be used for additional testing.

The agent we selected for this set of tests was an obvious choice. Constructed by one of us (Laird) to run with the latest version of Soar, it is suitably complex (employing ~600 rules) and operates in the highly dynamic and complex environment of the Quake II computer game. Although Quake II shares few, if any, attributes with the Towers of Hanoi puzzle, application of our evaluation methodology within this new domain was straightforward. As in Towers of Hanoi, a single set of knowledge could be used to test all of the Soar architectural variants, and testing followed the same basic procedure. The only significant difference resulted from the fact that in the Quake II environment, exogenous events are possible. Unless the world's events unfold in exactly the same manner between tests of two architectural variants, it is impossible to determine whether the agents interacting with the world underwent the same processes of reasoning. As a result, whether or not the performance of the architecture/agent pairs is comparable also depends on the ability to ensure that the world's events unfold in a repeatable manner.

To ensure that this did happen, the agent was initially allowed to operate in the Quake II environment by competing against a human opponent for a predetermined amount of time. During this phase, the agent's sensory inputs were recorded and stored in a file. During benchmarking, however, agents did not actually communicate with Quake II. Rather, their sensory input was replayed in exactly the same manner as occurred during the initial recording phase. Not only did this allow us to ensure that agents always performed the same reasoning, but because agent inputs were read from disk and stored in memory prior to benchmarking, it also guaranteed that timing results would reflect Soar's true performance, and not be skewed by a communication bottleneck with the environment.

Figure 1 shows the run time performance in Quake II for the standard version of Soar (variant 1) and Soar Lite (variant 3). The performance was measured by recording the time required to complete each of 380 successive decision cycles. The histogram in Figure 1 shows the number of cycles that were performed within specific time frames. The best behavior is to have all of the decision cycles execute in the minimal amount of time (to the left). As this behavior is difficult to achieve, a secondary goal is to have a low variance without any outliers so that there are no decisions that disrupt the overall system execution. In the figure, the standard version of Soar does have a high variance and many outliers. In contrast, the Soar Lite version shifts the histogram to the left so that almost all of the decisions execute in .03 seconds or less. There is one significant outlier at .08, but that is the first decision when working memory is initialized and it is irrelevant to the overall runtime performance of the bot. This illustrates that Soar Lite not only improves the aggregate execution time (in this case there is a factor of 3 improvement in average execution time) but improves it at the level of individual decisions in a such a way as to decrease the overall maximum computational requirements of any single decision.

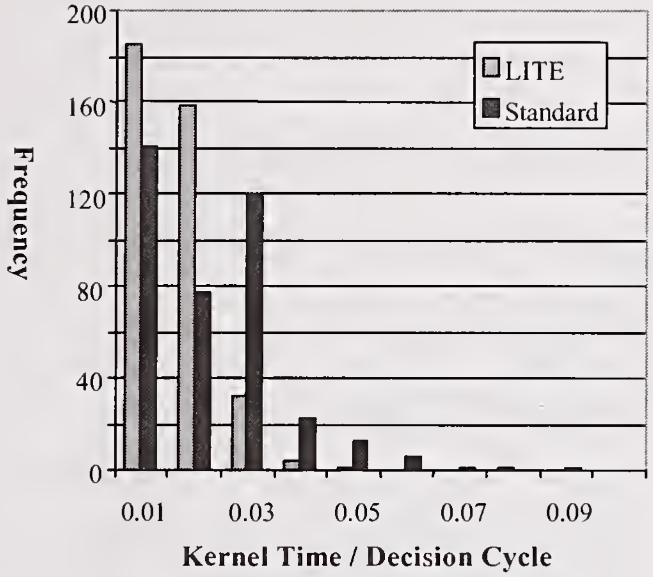


Figure 1. Execution cost in Quake II

8. Comparing Multiple Agent Architectures

The methodology described in section 3 and that we have employed to examine the performance of the Soar architecture and some of its variants can also be used to examine or compare two distinct architectures directly. The same steps are applied as outlined previously, but the architectures need not be split into modules. The most difficult aspects of using our method for distinct architectures are deciding what class of problems to examine and how to implement the agents. The difficulties stem from the fact that problem definitions must be highly constrained so that each agent's knowledge is extremely similar, if not identical. At the same time, however, these problem definitions are likely to require more flexibility than in the single architecture case, because perfect behavioral analogues may not exist between two architectures. Thus, the burden is on the research team to ensure that agents are appropriately similar and that they encode the same knowledge. As in the single architecture case, once agents have been created, their performance in the problem domain can be measured and compared.

8.1 The CLIPS Architecture

As an initial choice of a second architecture with which to conduct our evaluation, we have selected CLIPS [1]. Like Soar, CLIPS is a forward-chaining production system based on the RETE matching algorithm. In CLIPS, short term, potentially volatile, knowledge is stored in STM in the form of lists. Each list is given a name, or type, which is essentially the first element in that list. The remaining elements are labeled either explicitly or implicitly by referring to their position in the list. Each element is also a constant value, either numeric or string, and there is no architectural mechanism for referring to the contents of another list, or pointing to another slot.

As in Soar, long-term knowledge is stored as rules that are defined by the programmer. The conditions of these rules match against the contents of STM. Conditions can contain combinations of both constants and variables; however, variables may not be bound to list names or to slot labels. CLIPS, however, is not limited to using simple predicates in the right hand side of a rule as is Soar. A large variety of predefined predicates, as well as user defined predicates and functions can also be used as conditions. The action side of a CLIPS rule is used to modify the contents of STM or to execute external procedures.

CLIPS deploys knowledge via serial rule firings. The basic execution cycle consists of two steps:

- First, rule matches are calculated by comparing the conditions of each rule to the contents of STM.
- Second, successfully matched rules are placed into an ordered list such that the instantiated rule at the top of the list has highest priority.

Priority is defined using two methods. The first of these is a rule level conflict resolution mechanism called salience, which can either be a constant value, or a value calculated at run time. Rules with higher salience are placed higher in the list. In many cases, salience alone is not enough to determine a single highest priority rule. In these cases, CLIPS defers to one of a few user selected architectural mechanisms called search strategies, which orders rules of equal salience. At this point, the first rule in the list is fired and the entire process repeats itself. When no more rules are able to fire, the system halts.

Although there are many similarities between the Soar and CLIPS architectures, the differences are equally significant. These differences occur in each of the three architectural areas we have discussed: knowledge representation, knowledge deployment, and execution cycle. Recall for example, that Soar stores short-term knowledge in a directed graph structure and can perform variable binding on any slot in a memory element. CLIPS, on the other hand, stores short-term knowledge in lists, and cannot bind variables to the list name or to the names of its slots. Moreover Soar fires all matching rules in parallel whereas CLIPS fires only the highest priority rule. An additional difference is that Soar natively supports the decision making process within its execution cycle whereas CLIPS does not.

8.2 Towers of Hanoi revisited

We have examined CLIPS in the Towers of Hanoi domain using the same parameters that were used in our earlier evaluations of the Soar architecture. Note, however that the absolute timing data is not the same as in the first runs. Previous runs in this domain were done on different machines and measure total CPU time, not just Soar kernel time. Below, we briefly review the decision-making strategies of each agent and discuss the particularities of the CLIPS implementation. Notice that two additional categories have been added to further constrain the implementations and to examine areas that may be more amenable to the CLIPS architecture.

- Mutually Exclusive Reactions - Action conditions are mutually exclusive, and no symbol is declared to represent the action being pursued. In both Soar and CLIPS this is done by the construction of individual rules which specificity the preconditions of an action and its effects. Actions are applied sequentially within the world, and the programmer must ensure that no conflicts arise between two actions.

- Simple, Declared Actions - Similar to the first category, but in this case the action being pursued is declaratively represented. In Soar this is done using operators to represent the action. In CLIPS a fact is asserted which describes the current action being pursued. Once again however, the programmer must ensure that action preconditions are mutually exclusive.

- Two-Phase Decision - Two distinct phases are used to make the decision. In the first phase, actions are proposed via declarative symbolic representation. In the second phase one of these actions is selected and pursued. Note that this means that preferences corresponding to a specific action must be expressed simultaneous to the creation of the action symbol (e.g. within the same rule). In Soar, this is done using the architecturally supported decision phase, and the same rule is used both to propose an operator as to express its preference. In CLIPS, partitioning knowledge into a salience hierarchy supports the two phases. This guarantees that the first phase (action proposal) completes before the second phase (selection) begins.

- Three-Phase Decision - Three distinct phases are used to make the decision: proposal, preference and selection. These distinct phases help support situation dependent preference structures without an explosion of individual rules. In CLIPS this is done using a three-stage salience hierarchy.

- Goal Directed - High-level actions, possibly requiring more than one action to complete, are used to constrain rule matching. In CLIPS, goals are maintained declaratively and represented in a stack. Two Soar implementations were examined, one using Soar's native mechanism as demonstrated in the previous trials, and the other using a declarative stack similar to that used in the CLIPS implementation.

Figure 2 shows CLIPS and Soar performance in the Towers of Hanoi domain. Qualitatively, performance is very similar between the architectures except at the end points. On the left-hand side of the graph, the Soar agent performs markedly worse than the corresponding CLIPS agent. This performance difference can likely be explained by the fact that this Soar agent does not use the operator construct. As a result, it does not benefit nearly as much from constraining the rule matching as the other Soar agents do, and thus suffers an increase in execution time. At the other end of the graph, Soar and CLIPS behavior are once again divergent. In CLIPS we can attribute the performance increase to the fact that the problem's recursive nature allows the proper puzzle-solving knowledge to be easily expressed with a goal stack, and results in highly constrained rule matching. In the standard version of Soar (point 2), however, we have already seen that the benefits of subgoaling are dominated by the costs of Soar's architecturally supported subgoaling mechanism. However, when performance is re-examined using architecturally supported subgoals in the Soar-Lite variant (point 2') or when using a declarative subgoal stack similar in nature to the CLIPS implementation (point 1) the difference between the Soar and CLIPS agent's performance is minimal. In all, the similarity of performance between the declarative goal stack implementations in both Soar and CLIPS, and the architectural implementation in Soar-Lite, indicate that in simple environments such as Towers of Hanoi, declarative subgoaling provides a simple and efficient means of problem solving. As tasks become increasingly complex, however, we expect that the rule-based techniques employed by these implementations will become significantly less efficient than the lightweight architectural counterpart of Soar-Lite

9. Related Work

Examining differences between agent architectures has received relatively little attention compared to the complementary task of examining how different agent strategies are more or less suited to a particular problem. Nonetheless, a variety of approaches have appeared in the literature. The majority of architectural evaluations can be placed into a single group that we refer to as categorical comparisons [5,8,9,14]. Within this body of work, architectures are evaluated at a high level, in a domain-independent manner, typically based on whether they natively support certain features (e.g. backward or forward chaining, or the ability to make real-time commitments). The benefits of this approach are that the concise tabular data, representative of these studies, may allow architectures to be quickly assessed as having or not having the minimal necessary capabilities to perform the task at hand. Categorical evaluations are most useful when they examine aspects of the architecture that are extremely difficult, or impossible, to emulate using additional, programmer supplied, knowledge.

However, the high-level approach of categorical evaluations can also be a short-coming, In particular, the many situations in which architectural features can, in fact, be successfully emulated with addition knowledge are often not explored. More over, because these studies rarely incorporate benchmarks, there is often no indication as to the relative performance of different architectures or their underlying features.

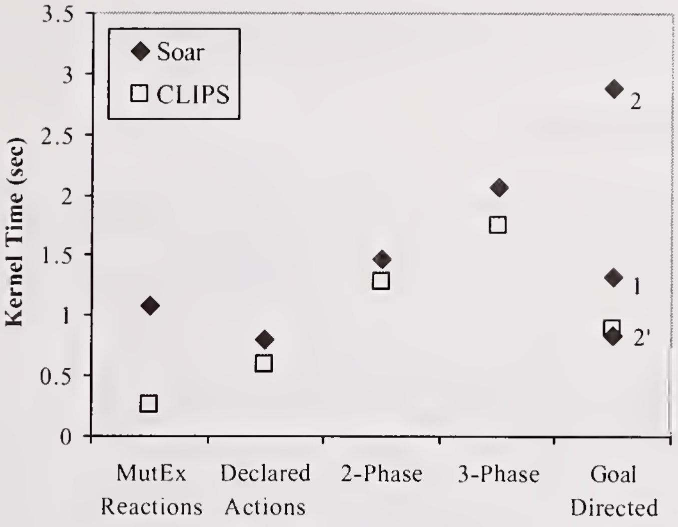


Figure 2. Execution Time of Soar and CLIPS in Towers of Hanoi

In contrast to high-level categorical comparisons, the Sisyphus-VT initiative examined the problem of implementing a complex real-world problem on a number of different architectures [13]. Although the pursuit of complex, real world, problems as test bed domains is a laudable undertaking, the implementation overhead is extremely high. As a result, independent teams of programmers, expert in one particular architecture, carried out the implementations. A critical difference between the methodologies used in the Sisyphus-VT study, and the one we have presented is that we emphasize that the problem solving methods used by two comparable agents should be strictly specified and adhered to. Sisyphus-VT, on the other hand, allowed relative freedom in this area. Although this freedom allows programmers to use a problem solving method which they feel is best suited to their architecture, it also means that differences in two agents' performances might be attributable more to differences in their knowledge, than to differences between the architecture which serve as their foundations. Plant and Salinas attempted to circumvent the problem of confounding the contribution of knowledge and architecture to the overall performance rating in their 1994 study [11]. Under their methodology, agents were constructed in a generic manner so that they had minimal reliance on architecturally specific constructs. This allowed them to create agents for each architecture based primarily on syntactic transformation of a single, handcrafted, specification. This methodology certainly adheres to our requirement of strictly specifying the agent's underlying problem solving methods. But it deviates from our requirements because it does not examine a range of these underlying methods. As a result, it is less likely that the benchmarks will include near-optimal implementations for any architecture, especially since reliance on architecturally specific constructs is purposely minimized.

10. Discussion

The methodology we have presented allows the performance of two architectures, as well as variations of a single architecture to be compared directly. Our methodology is an evolution of prior research, and emphasizes aspects of the benchmark design (e.g. problem-solving specification), which help ensure that agents built using two different architectures use equivalent knowledge. An initial application of our comparative approach has shown significant differences between 3 variations of the standard Soar architecture when Soar's learning capabilities are not required. This hypothesis was further supported by examining the performance of an agent in the complex, real-world, environment of Quake II. The broader implication of this finding is that knowledge both about the domain and about the implementation of the agent should play a role in deciding what architecture (and what architectural features) are most suitable for a particular circumstance.

We have also shown that the same methodology used to compare variations of a single architecture can also be used to compare two distinct architectures. We have illustrated this application with an initial comparison of Soar and CLIPS. Results from this set of tests indicated both qualitative and quantitative differences in their performance, and have also illustrated the potential performance savings that can be achieved by an architecture whose features are well suited to the current task.

We believe that the work presented in this paper provides a good foundation for addressing the question of what are the resource requirements of architectural properties, or, which properties of an architecture are most suitable for a given situation. Because the needs of intelligent agents often simultaneously push architectures to support a wide array of features and to be highly efficient in terms of run-time performance, an improved understanding of the answers to these basic questions is important.

11. Acknowledgments

This research was funded in part by grant N61339-99-C-0104 from ONR and NAWCTSD.

12. References

[1] CLIPS Reference Manual: Version 6.05.

[2] R. B. Doorenbos. *Production Matching for Large Learning Systems*. PhD thesis, Carnegie Mellon University, 1995.

[3] C. L. Forgy. *On the Efficient Implementation of Production Systems*. PhD thesis, Carnegie Mellon University, 1979.

[4] E. Gat. Integrating planning and reacting in a heterogeneous asynchronous architecture for mobile robots. In Proceedings of the Tenth National Conference on Artificial Intelligence, pp. 809-15.

[5] W. B. Gevarter. The nature and evaluation of commercial expert system building tools. *Computer*, 20(5):24-41, 1987.

[6] F. F. Ingrand, M. P. Georgeff, and A. S. Rao. An architecture for real-time reasoning and system control. *IEEE Expert*, 7(6):34-44, Dec. 1992.

[7] J. E. Laird, A. Newell, and P.S. Rosenbloom. Soar: An architecture for general intelligence. *Artificial Intelligence*, 1987.

[8] J. Lee and S. I. Yoo. Reactive-system approaches to agent architectures. In N.R. Jennings and Y. Lesperance, editors, Intelligent Agents VI - Proceedings of the Sixth International Workshop on Agent Theories, Architectures, and Languages (ATAL-99), Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin, 2000.

[9] W. A. Mettrey. A comparative evaluation of expert system tools. *Computer*, 24(2): 19-31, 1991.

[10] A. Newell. *Unified Theories of Cognition*. Harvard University Press, Cambridge, MA, 1990.

[11] R. T. Plant and J. P. Salinas. Expert system shell benchmarks: The missing comparison factor. *Information & Management*, 27:89-101, 1994.

[12] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*, chapter 2, pages 31-52. Prentice-Hall, Upper Saddle River, NJ, 1995.

[13] A. Th. Schreiber and W. P. Birmingham, Editorial: the Sisyphus-VT initiative. *International Journal of Human-Computer Studies*, 44(3): 275-280, 1996.

[14] A. C. Stylianou, R.D. Smith, and G. R. Madey. An empirical model for the evaluation and selection of expert system shells. *Expert Systems With Applications*, 8(1): 143-155, 1995.

[15] S. A. Wallace and J. E. Laird, Toward a methodology for AI architecture evaluation: comparing Soar and CLIPS, in N.R. Jennings and Y. Lesperance, editors, Intelligent Agents VI - Proceedings of the Sixth International Workshop on Agent Theories, Architectures, and Languages (ATAL-99), Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin, 2000.

[16] S. A. Wallace, J. E. Laird and K. J. Coulter. Examining the resource requirements of artificial intelligence architectures. In Proceedings of the Ninth Conference on Computer Generated Forces and Behavioral Representation , pp. 73-83

A Metric For Monitoring And Retaining Flight Software Performance

Chariya Peterson
Computer Sciences Corp. 7700 Hubble Drive,
Lanham-Seabrook, MD 20706, cpeters5@csc.com

ABSTRACT

The control of spacecraft dynamics are handled by on board flight software which are typically based on sequential algorithm such as Extended Kalman filter (EKF) to perform closed-loop automatic control. This level of automation does not require any decision-making or learning capability. Decision-making capability comes to play at the higher level of autonomy in task management, such as mode/model selection, planning and scheduling. Most of these functions are still performed on ground and are not fully autonomous. In this paper, we propose the concept of intelligent flight software that is capable of learning, and improving its performance in the future based on information gained in the past. This capability will enable the software to appropriately deal with uncertainty or incomplete knowledge of model or environment. To be precise, we will focus on on-board Attitiude Determination and Control software (ADCS). A typical ADCS is an automation where attitude solutions are computed dynamically via a filter and controlled by a closed-loop PID process. The performance of a typical ADCS is maintained by Flight Dynamics ground personnel. These tasks involve, among other things, attitude determination and validation, and attitude sensor model calibration. In this paper, we propose an intelligent ADCS that is able to monitor its own performance and able to perform a self-calibration when needed.

KEYWORD: *Control Theory, Intelligent software, Uncertainty Management, Machine Learning*

1. INTRODUCTION

The task of maintaining long-term performance and accuracy of software onboard a spacecraft can be a major factor in the cost of operations. In particular, the control and maintenance of constellation or distributed spacecraft undoubtedly pose a great challenge, since the complexity of multiple spacecraft flying in formation grows rapidly as the number of spacecraft in the formation increases. Eventually, new approaches will be required in developing viable control systems that can handle the complexity of the data and that are flexible, reliable and efficient. These new approaches will have to face the problems that are encountered during the development of a control system, in particular how to deal with uncertainties in the application domain and how to balance between efficiency and complexity of the system. The accuracy of control software depends on how much information about the domain is modeled into the system. The more information taken into account, the more complex the system becomes, leading to higher computational cost. Hence pure model-based approaches will undoubtedly be too costly for a large control system.

Most of the material covered in this paper is in the paper presented at the SpaceOps Symposium, Toulouse, France June 19-23, 2000 [7].

2. SOFTWARE PERFORMANCE

Typical flight software performs closed-loop automation control without any high level decision-making, or learning involved. On the other hand, autonomy are added to the flight software in terms of flight or ground component that aims to increase operational range of the software, involving model selection, performance monitoring and self-calibration and tuning. We identify the intelligence of modern flight software with its decision-making capability, which results in the autonomy level of the software. We measure the intelligence of onboard software in terms of its ability to learn from experience and its rate of success. In the lowest level, we define the performance of flight software as a measure of the closeness between the observed and the predicted state of the systems. These quantities are usually referred to as *residuals*. Understanding the uncertainty underlying these residuals, identifying their controlling factors, and quantifying the propagation of these factors through the model for the system can lead to an improvement in the intelligence of the software.

On-board ADCS generally reacts directly with input sensor measurements and thruster control via simple closed-loop process. The typical operational range of such standard ADCS is narrow, and as a result, the system may perform poorly under uncertain conditions such as incomplete knowledge of world model, or unanticipated changes in the environment. To cope with this problem, the models used in the software are parameterized. The model parameters are adjusted regularly to maintain the accuracy level of the software. These tasks are typically performed manually on the ground in a regular basis. This suggests that, the intelligence of flight software may be increased by enable the software with self-monitoring and self-calibration functionality. Recently, there have been a few research efforts in increasing the intelligence of flight software: for instance, the Remote Agent Experiment (RAX) onboard DS-1 spacecraft [1], and autonomous on-board dynamic monitoring developed at Jet Propulsion Lab, [BEAM].

We propose to develop the Monitoring and Autonomous Self-Tuning (MAST) system that aims to maintain the efficiency of onboard software by dealing with uncertainty in an appropriate way. MAST is an extension of a project at NASA/Goddard Space Flight Center (GSFC): Autonomous Model-based Trend Analysis System (AMTAS) [2]. MAST extends the objective of ASCAL from health and safety management of hardware to dynamic applications. MAST uses machine learning approach to handle uncertainty in the problem domain, resulting in the reduction of over all computational complexity. The underlying concept of this technique is a reinforcement learning scheme based on cumulative probability generated by the past performance of the system. The success of MAST depends largely on the reinforcement scheme used in the tuning algorithm and its ability to remember and learn from its experience.

3. THE MONITORING PROCESS

MAST consists of two main parts: a *monitor* and a *tuner*. The monitor is a real-time dynamic system that monitors relevant residual output of the software it is monitoring. The step size of the sampling time varies depending on the parameters being monitored. The state of the monitor is the quantity representing software performance in real time. When the state of the monitor approaches a given threshold, the tuning process will be initiated. This process has no intelligence i.e. it does not require any decision-making capability.

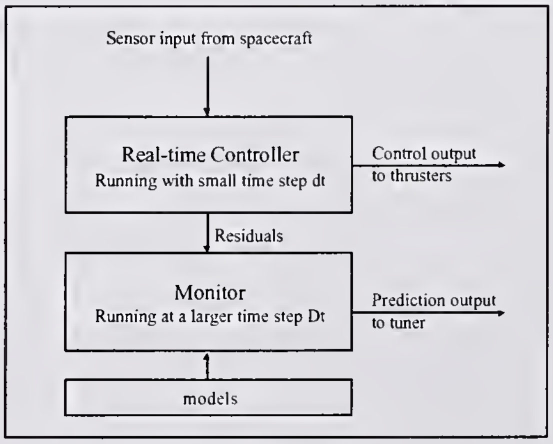


Figure 1. Monitoring Mode

Figure 1 demonstrates the monitoring mode, which consists of the software being monitored and a monitor, both running in real time. The detail description of the monitor depends on the software being monitored. It is necessary that the monitor have sufficient knowledge of the software in order to make an accurate prediction and diagnosis of the problems.

For intelligent ADCS where Kalman filter is used for real-time computation of attitude solutions, the performance of ADCS is monitored by trending the attitude solutions and the effective sensor measurements. In a nutshell, attitude of spacecraft is continuously propagated through time using angular rate measurements from gyroscopes. These attitude solutions are not very accurate since gyroscope measurements are usually erroneous. The accuracy of attitude computation can be improved by occasionally comparing the attitude with vector (directional) measurements from available sensors on-board such as star trackers, magnetometers, sun or earth sensors. Kalman filter is an algorithm that performs such sequential process of propagating and mcasurement updating. The size of the residuals of sensor measurements reflects the performance of ADCS.

Let x denotes the state vector estimated by the software and s denotes the vector of sensor parameters being monitored and calibrated. Assume that an expected state vector x_a is given. x_a may be obtained in various ways depending on the software and on sensors and parameters being monitored. Let the software be driven by the dynamic system

$$\begin{aligned} \dot{x}(t) &= f(x(t)) + u(t) \\ z_{r,k} &= G(s_r, x(t_k)) + w(t_k) \end{aligned} \qquad (1)$$

where $z_{r,k}$ is the measurement for sensor r at time t_k, and s_r is the parameter vector associated with the model of measurement r. The process noise u and measurement noise w is assumed to be uncorrelated white Gaussian noise with zero mean. During the normal mode of operation s_r are kept constant. The performance of (1) is observable from the deviation of certain quantities, such as state residuals $x - x_a$, and sensor residuals, $z_{r,k} - G(s_r, x_a(t_k))$. Let ξ represents the vector of the desired residual observations. The monitoring process is then defined via a tracking process, i.e. the linear dynamic of ξ and its slope $\dot{\xi}$:

$$\xi(t_{K+1}) = \xi(t_K) + \Delta t \cdot \dot{\xi}(t_K) + \tfrac{1}{2}\Delta t^2 v(t_K)$$

$$\dot{\xi}(t_{K+1}) = \dot{\xi}(t_K) + \Delta t \cdot v(t_K)$$

where V is a zero mean white Gaussian uncorrelated acceleration noise. The time step $\Delta t = t_{K+1} - t_K$ for rcsidual samplings may be larger than the time step of the input system (1). Let $\hat{x} = [\xi \;\; \dot{\xi}]'$. Then the state-space representation of the predictor can be written as

$$\begin{aligned} \hat{x}(t_{K+1}) &= A\cdot\hat{x}(t_K) + V\cdot v(t_{K+1}) \\ \hat{z}_K &= H\cdot\hat{x}(t_K) + \gamma_k \end{aligned} \qquad (2)$$

where

$$A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix},\ V = [\Delta t^2/2 \;\; \Delta t],\ H = [1 \;\; 0].$$

Note that, the measurement $\hat{z}_K$ represents the residual sampling while the state $\hat{x}(t_K)$ measures the level of performance of (1) during the time t_K. A propagation of $\hat{x}(t_K)$ predicts if and when the performance of (1) approaches an acceptable threshold at a ccrtain time in the future. The system (1) and the predictor (2) connect as shown in Figure 1. Higher order derivatives of state residuals can also be included in $\hat{x}(t_K)$ in a sirnilar way. In which case, we would have a higher order predictor. Higher order derivative may be crucial for software systems that are sensitive to uncertainties in measurement models, which is generally the case for a highly non-linear, chaotic or unstable systems.

4. TUNING PROCESS

The tuning process is a closed-loop learning algorithm based on a reinforcement learning scheme. The goal of the tuning process is to restore the performance of the software by iteratively adjusting relevant model parameters in a "*certain way*" until a cost function is minimized. The tuner possesses two types of intelligence:

1) During each cycle the tuner will select which parameter to adjust. This selection is MAST's long-term knowledge on its past tuning experience. This intelligence is measured by the rate of success in software tuning.
2) The amount of adjustment for each parameter. This selection is a short-term knowledge generated by the reinforcement scheme of the learning algorithm. This type of intelligence is measured by the rate of convergence for each particular tuning process.

Note that, the learning approach does not give an optimal solution, but it has a much wider operational range than the conventional optimal batch least square or filter techniques. This is simply because; MAST automatically accumulate and reuse its past activities in its long-term memory, which will enable the system to react and adapt to changes in the environment. This approach is therefore appropriate for problems with large degree of uncertainties. Moreover, this technique is not critically dependent on the detailed knowledge of the software being tuned. As a result, some of the technical restrictions generally required in conventional techniques such as linearity, or conditions on process and mcasurement noises are not required if a learning algorithm is used. It should be noted that the tuner is an off-line algorithm, or a process running in parallel and isolated from the routine operation of the software. Not until the tuning goal has been reached, that the software will be updated with the new values for the model parameters. Hence, the tuner may be performed on the ground or on an onboard computer.

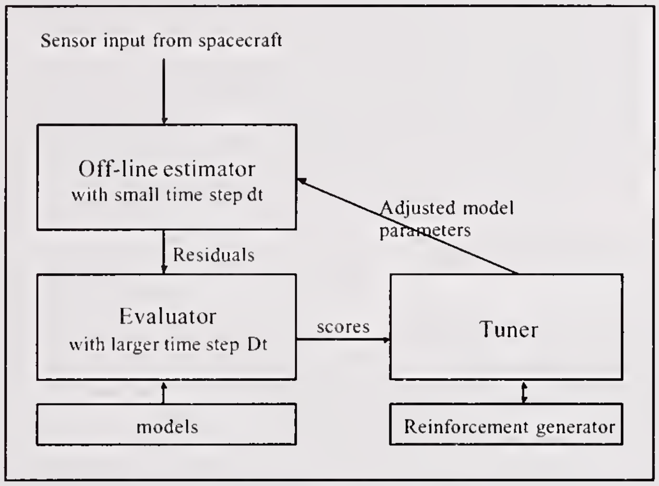


Figure 2. Tuning Mode

Figure 2 demonstrates the tuning mode. In this mode, there are three components connected in a closed-loop: an off-line copy of the software being monitored, the evaluator, and the tuner. The evaluator measures the convergence of the tuning solutions and the tuner makes appropriate adjustment to certain model parameters of the software guided by a reinforcement learning scheme, generated by an uncertainty handling technique. Several techniques have been used by various research projects in reinforcement learning. In MAST, the scheme is based on the Local Dempster-Shafer theory (LDS) which is a modification of the Dempster-Shafer theory of belief and evidence [4,5]. For the detail description of LDS we refer to [2,6]. LDS is specifically developed to deal with systems with large number of variables. As opposed to the monitor, the evaluator and the tuner are generic processes that do not require detailed knowledge of the software being tuned. Their basic requirements are a set of software parameters to be tuned and an appropriate cost function that models the inaccuracies of the software. The evaluator evaluates and scores the result of each cycle by examining the effect of the parameter adjustment on the cost function. Based on this score, the tuner continues to adjust the parameters until the process converges.

Reinforcement learning is the type of learning that is popular among most current researches in machine learning and statistical pattern recognition. Other popular type of learning systems such as artificial neural network, requires *a priori* training from examples provided by an experienced supervisor. Such systems are not quite appropriate for problems involving learning from interaction. In interactive problems it is often impractical to obtain examples of desired behavior ahead of time, which are both correct and representative of all the situations to which the system has to react. In an unknown situation, where learning is most beneficial, the system must be able to learn proactively from its own experience.

During the tuning process, the parameter adjustment is based on the rate of convergence (or divergence) of the residuals during the previous two (or more) cycles. Assume there are n sensor parameters to be adjusted, i.e. the dimension of the parameter vector p_K is n. The parameters can be increased or decreased by Δp_K. The set H of all possible adjustments has $\sum_{i=0}^{n} 2^i \binom{n}{i}$ elements. Each element is a set of parameters with a plus (+) or minus (–) sign to denote if the parameter is being increased or decreased. For instance, an increase in parameter a and a decrease in parameter b is represented by the "signed" set $\{a_+, b_-\}$. During each loop K, the step size Δp_K is computed, and the set H is constructed. An indexed by a cumulative probability distribution p_K which generated by LDS theory. The learning process in the tuner is precisely the mechanism that adapts p_K to obtain the new index p_{K+1} for the next cycle. The original Dempster-Shafer theory is defined on a set of n elements. Recall that, H is a set of all possible ways of modifying model parameters being tuned. A mass function on H is a probability function that assigns a degree of belief to each of its element. The mass function satisfies the following conditions

$$\sum_{A\supseteq H} m(A) \quad = 1, \qquad \text{for } A \neq \emptyset \quad \text{and}$$

$$m(\emptyset) \quad = 0$$

Two mass functions m_1 and m_2 on H can be combined into a single mass function $m_1 \otimes m_2$ by the Dempster composition rule:

$$m_1 \otimes m_2(A) \quad = \quad \sum_{B\cup C=A} m_1(B)m_2(C) \Big/ (1- \sum_{B\cup C=\emptyset} m_1(B)m_2(C))$$

for $A \neq \emptyset$, $\quad m_1 \otimes m_2(\emptyset) = 0$.

These mass functions are used to generate the degree of belief associated to each element of H. A belief function generated by a mass function m is defined as:

$$p : H \to [0,1];\ b(A) = \sum_{B\supseteq A} m(B)$$

where the union between two signed sets is obtained by "adding" all elements in the two sets according to their sign. This way, every subset of the form $\{a_+, a_-\}$ will all be cancelled out. In statistical terms, the belief function is a cumulative probability on H.

During a tuning cycle K, the belief function p_K is evaluated and used as an index set for H. If the resulting residuals are found to decrease with a faster rate or increase with a lower rate, the tuner will re-compute the next belief vector p_{K+1} by applying a positive learning algorithm described in [1,9]. The new index will strengthen the performance of the cycle K. Conversely, if the residuals performed in the negative manner, then the negative learning algorithm will be applied, resulting in lessening the degree of belief on the failed action.

The learning process discussed above is the simplest application of the (modified) DS theory to the tuner. In practice this algorithm can be enhanced in various ways to increase the performance and robustness of the tuner. First, the localization of the DS theory on H defined in [1,9] will reduce the size of search space. Second, the size of parameter increment may be decreased as the residuals begin to converge. Third, the use of hierarchical or multilevel learning systems accelerates the learning process (more so for the initial rate of learning) and simplifies the structure of the tuner in each layer.

5. TWO APPLICATIONS OF MAST

The attitude monitoring and self-calibration (ASCAL) [3], and the maintenance of spacecraft formation. In the first application, the accuracy of attitude software depends on, among other things, the accuracy of sensor models. These models are generally a function with parameters representing relevant uncertainties such as bias, scale factor or misalignment. In the beginning, these parameters are set at certain pre-calibrated values and are manually tuned and updated periodically throughout the life of the spacecraft. Some tuning processes are routine activities, while others are elaborated and performed on ground by attitude specialists. In this proposed application, MAST will automatically monitor and tune a set of sensor parameters.

The second example is the maintenance of large formation of spacecraft. The task of controlling a number of spacecraft to fly in formation is more complicated than controlling a single spacecraft. One problem that may be encountered in the development of formation control algorithms for large formation is the complexity that arises from the high degree of freedom of the system. In practice, the conventional approach based on state-space representation is manageable only for formation of a small number (2-3) of spacecraft. The complexity increases in a large formation, which makes the control algorithm computationally intensive. Moreover, uncertainties in the system models or from environmental disturbances can be propagated and magnified. To correct these errors the control system has to be tuned often and regularly to keep the formation intact by continuously monitoring and adjusting the position of each individual spacecraft. Ideally, these tasks should be performed onboard, and hence efficient and fast algorithms for the real-time solution of such a large-scale optimization problem are needed.

6. REFERENCES

[1] N. Muscettola, B. Smith, C. Fry, S. Chien, K. Rajan, G. Rabideau, and D. Yan, *On-Borad Planning for New Millennium Deep Space One Autonomy,* Proceedings of the IEEE Aerospace Conference, Aspen, CO 1997.

[2] C. Sary, C. Peterson, J. Rowe, K. Mueller, W. Truszkowski, T. Ames, N. Ziyad, *Automated Multimodal Trend Analysis System*, Proceedings of the AAAI Spring Symposium 1998.

[3] C. Peterson, J. Rowe, K. Mueller, N. Ziyad, *ASCAL: Autonomous Attitude Sensor Calibration*, proceedings of the Flight Mechanics Symposium, NASA/GSFC May 1999.M. D. Shuster, D. M. Chitre, and D. P. Niebur, *Inflight Estimation of Spacecraft Attitude Sensor Accuracies and Alignments*, J. of Guidance and Control, 5, 4, 1982.

[4] G. A. Shafer, *Mathematical Theory of Evidence*, Princeton U. Press, 1976.

[5] A. P. Dempster, *Upper and Lower Probabilities Induced by Multivalued Mappings*, Annals of Math. Stat. 38, pp. 325-329, 1967.

[6] C. Peterson, *Local Dempster Shafer Theory*, CSC-AMTAS-98001, C.S.C Internal Report

[7] C. Peterson and N. Ziyad, *Autonomous Performance Monitoring System: Monitoring and Self-Tuning (MAST),* Proceedings of the SpaceOps 2000 Symposium, Toulouse, France, June 19-23.

Flexible Robotic Assembly

David P. Gravel
Senior Technical Specialist
Ford Advanced Manufacturing Technology Development
Detroit, MI 48239

Wyatt S. Newman
Professor EECS Dept., Case Western Reserve University
Cleveland, OH 44106

ABSTRACT

Application of robots in automobile manufacturing plants is primarily limited to material handling and spot welding operations. Only 3 percent of all robotic applications are currently performing tasks related to assembly [1]. These assembly tasks often have the characteristic that the positional uncertainties in parts to be mated exceed the assembly tolerances by many times. Humans are particularly capable of part assembly under these conditions for three reasons: (1) Humans can apply compliant, goal directed forces and positions to the assembly task. (2) Humans have a powerful vision capacity that integrates well in the application of these goal directed forces and positions. (3) Humans are also quick to learn new assembly techniques and can often perform complex assembly tasks easily after a short training period.

This paper will detail the work done at Case Western Reserve University (CWRU) in Cleveland, OH, and by Perceptron in Plymouth, MI on a NIST ATP for Flexible Robotic Assembly for Powertrain Applications (FRAPA). FRAPA is a Joint Venture comprised of the following companies: Ford Motor Co., MicroDexterity Systems (MDS), ComauPICO, Perceptron, and the National Center for Manuf. Sciences. Also participating as subcontractors in the FRAPA project are Sandia National Laboratories, the University of Michigan, and CWRU.

This work will lead to the production of an autonomous system that exhibits all three characteristics that humans are naturally endowed with, that is: goal directed compliant forces and positions applied to part assembly, visual feedback to reduce the part location uncertainty, and learning algorithms that improve the performance of the assembly task over time.

STATUS OF CURRENT TECHNOLOGY: POSITION CONTROLLED ROBOTS

Robots used as position-servoed mechanisms are ineffective as an assembly tool in cases where the assembly tolerance is less than the positional uncertainty. This can be illustrated using a simple example of a peg-in-hole problem.

Suppose a position-controlled robot attempts to assemble a peg into a hole, but the hole position is not precisely controlled, and further, the assembly tolerance is comparatively small. Unless the center-lines of the peg and the hole are nearly parallel and lie closer together than the assembly tolerance, the robot will not be able to insert the peg into the hole. In a misaligned state, if the robot does attempt to perform the assembly, unacceptably high contact forces will be generated as the robot attempts to push the peg into place. Even if the peg is chamfered, a misalignment between the peg and hole would produce large side loading forces as the robot struggles to move the peg along the programmed centerline of insertion. This misalignment of the peg is likely to cause a jam if no means for compliance, such as a remote center compliance (RCC) device, is employed (see e.g [2]).

In practice, the geometry of real parts is often more complex than a peg and hole which, can further complicate robotic assembly and decrease the chances for a successful automated assembly using position-controlled robots.

I. A NEW ROBOT CONTROL PARADIGM

Wyatt Newman at Case Western Reserve University in Cleveland, OH has implemented a robotic control strategy called "Natural Admittance Control" (NAC) [3,4,5], on several robot platforms. Instead of having the robot stiffly track a precise trajectory, it is guided by an attractor point with pre-programmed stimulus-response dynamics. This new robot control strategy provides excellent rejection of Coulomb friction, good force responsiveness, and guaranteed contact stability.

A programmed point called the "attractor point" controls the direction of the applied force. It is the attractor point that is manipulated by the trajectory generator in the robot. The attractor point pulls the robot tool center point (TCP) towards it with strength proportional to programmable spring constants in the X, Y, Z, Y, P, R directions. The damping of the robot TCP is accomplished in a similar fashion with programmable damping coefficients Bx, By, Bz, By, Bp, and Br that model a damping forces proportional to the velocity components in the Vx, Vy, Vz, Vy, Vp, and Vr directions.

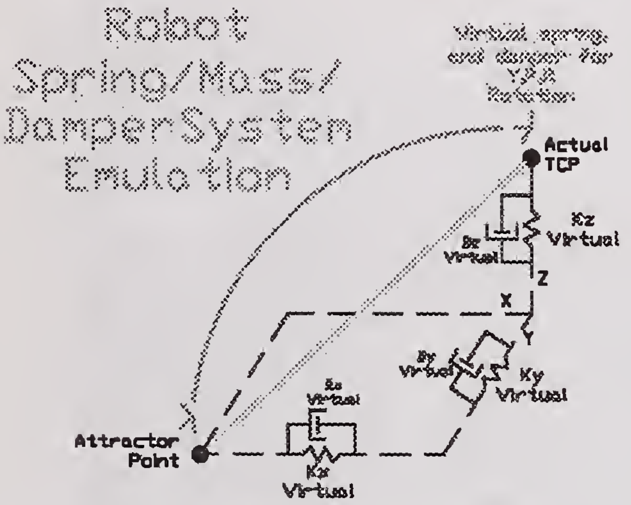


Fig. 1 Simplified schematic of robot TCP modeled as a programmable spring/mass/damper system.

Changes in the state of assembly can be accomplished by either altering the attractor point position or by changing the virtual spring or damping coefficients. In this manner it is possible to change the applied forces or endpoint dynamics as required for the assembly task.

In addition to the NAC robot control scheme, it is desirable to implement force controlled robotics on a robot platform that has low inertia, low gear friction, and high mechanical stiffness and is backdrivable. To address these control concerns and to provide a high fidelity response to input forces, a robot called Paradex has been created.

Paradex is a parallel robot structure comprised of six linear motor axes joined at a common distal platform. This configuration of robot meets the criterion for low inertia, excellent mechanical stiffness, and back drivability. The first generation of Paradex, shown in Fig. 2, is currently undergoing testing and development at CWRU.

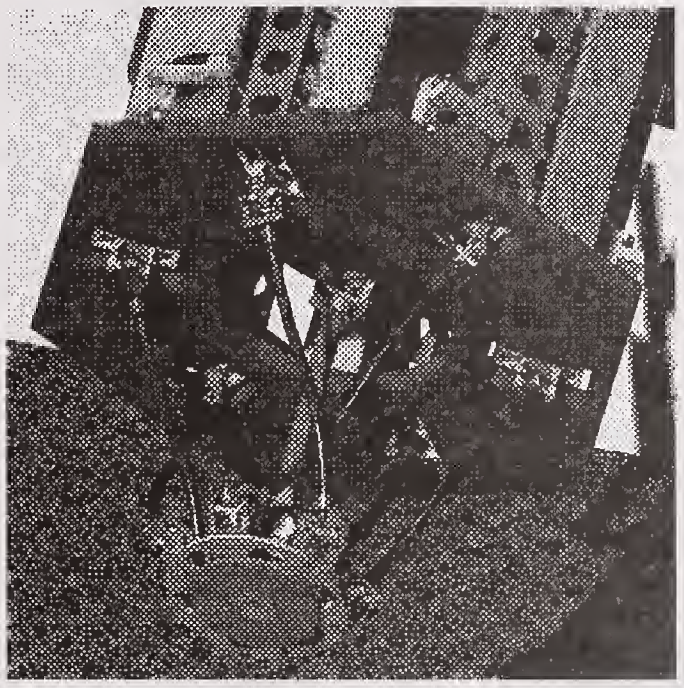


Fig. 2 FRAPA Paradex I robot

The NAC paradigm of robot control can be made to act as a conventional position controller by using very stiff virtual spring coefficients, e.g. for handling parts that are precisely or for spot-welding applications.

Given that NAC can be accomplished on a responsive robotic platform, the remaining task is to provide programmed strategies based on machine assembly states that can be first identified and then resolved by intelligently changing the virtual parameters and endpoint dynamics. This will be further discussed in section III.

NAC provides robots with the first human attribute of actively controlled, goal directed forces and positions being provided to parts to be assembled.

II. MACHINE VISION

Humans can easily visually locate and acquire parts to be assembled. Machines however, do not possess such a facility. Machine vision has historically been plagued by a lack of robustness in factory applications. Many of the problems related to the robustness of grayscale vision processing are due to a lack of contrast of features of interest, and to changes in lighting that cause object segmentation errors in the image.

3-D range imaging technology does not suffer from the contrast and lighting issues of grayscale vision processing, but many of the algorithms that can be commonly found for grayscale vision processing do not operate on 3-D range data. Under a NIST ATP for Flexible Robotic Assembly for Powertrain Applications (FRAPA), Perceptron in Plymouth, MI is developing new vision processing algorithms that will provide a robot with part pose information that a can be used to acquire randomly located parts.

One disadvantage of range imaging machine vision systems, however, is that the speed of image acquisition is about 2 s compared to 0.016 s for a CCD grayscale camera. This limits 3-D range imaging to applications to ones where the scene is static or slowly changing in time. Fortunately, in automotive applications, there are many examples of static scenes that this technology can be utilized.

Fig. 3 Example of parts delivered to an imprecise location that need to be acquired prior to assembly. Currently most of these operations are done manually in automotive plants.

Another disadvantage of range images is that distances measured from 0 -: show a periodicity. This results from the measured phase shift reflected from an object back repeating at a distance called an "ambiguity interval" due to the phase changing from 0 to 360 deg. and starting over at 0 deg. again. Fortunately, the ambiguity interval is somewhat large (~2m) so there are a number of plant applications that satisfy the condition of using a single ambiguity interval over the workspace.

The raw sensor data from the range vision system is acquired in a spherical coordinate frame with more pixels spatially located in the central region of the image than on the edge. An image region of interest (ROI) is rectified into a regularly spaced (X,Y,Z) 3-D array and processed by a new class of operators [6] that can quickly perform part identification and deliver part pose information as well.

These new vision operators are model based and yield a set of scalar values that represent both the existence of an object and pose information about the object. These operators are also relatively fast compared to correlation techniques and have full 6 degrees of freedom (DOF) isometry invariance.

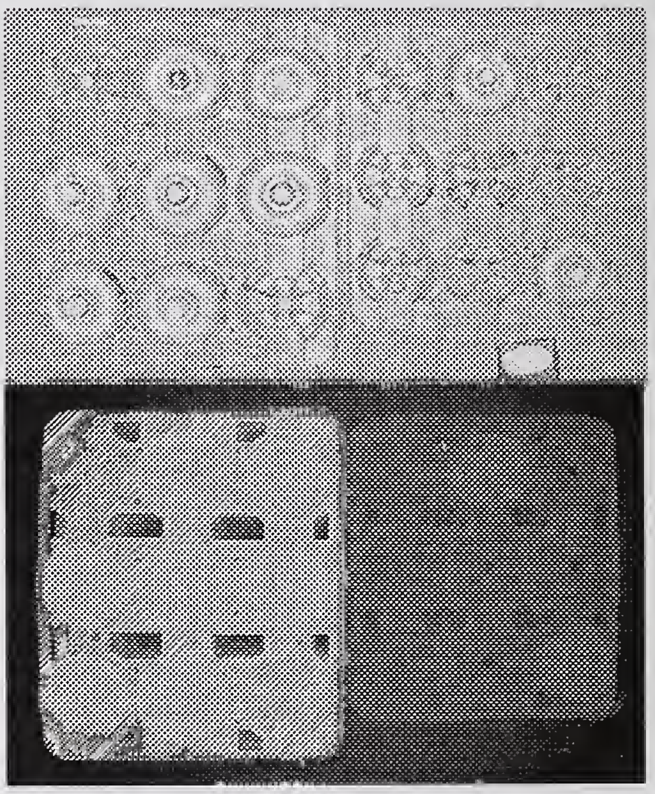


Fig. 4 shows a comparison of an intensity image(top) and a range image(bottom). These two pallets contain automotive torque converters. The bottom left hand pallet is a brighter image because it is closer in the range image.

The methodology employed by Perceptron to locate parts in roughly placed pallets follows these steps. (1) Locate the boundaries of the pallet and calculate ROI windows to isolate individual parts. (2) In each ROI window transform the row-column-range raw data into a uniformly spaced X-Y-Z representation. (3) Use the new vision operators to locate the part pose in each ROI. (4) Calculate robot coordinates in the robot world coordinate frame so the robot can pick up the parts from the pallet.

The capacity to locate roughly placed parts in 3-D will be used to robotically acquire these parts. This capacity provides automation with the second human attribute of coordinated vision to perform robotic assembly.

III. MACHINE LEARNING APPLIED TO ROBOTIC ASSEMBLY

Humans are able to learn complex assembly tasks relatively quickly. Machines, however, are notoriously difficult to "teach". Any ability to provide a machine with a learning capacity to perform a task will require decomposing the task into subtasks that can be understood and modeled in ways a machine can use.

Work at CWRU on machine learning of automotive assembly tasks has focused on four areas. (1) Gain an understanding how people perceive edges and boundaries when attempting to mate two parts together. (2) Understand strategies people invoke under various force feedback conditions (3) Create a model based system that uses signal feedback to direct the robot motion to the proper position for assembly (4) Implement the knowledge gained in steps 1-3 into a neural net robot controller.

At CWRU an idealized peg-in-hole virtual model was created for round and square peg-in-hole applications. Moments can be shown to exist as a portion of a peg passes over the region of a hole. A peg-in-hole model is similar to a simplified set of problems in transmission assembly. The goal is to extend this work to automotive assembly tasks such as a sun gear insertion into a planetary gear set.

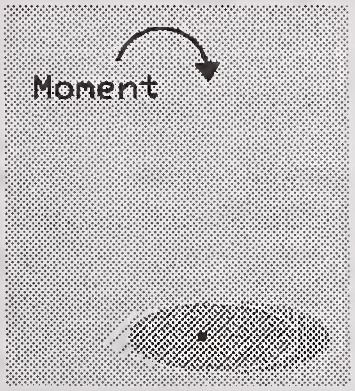


Fig. 5 Near the hole moment information exists that be exploited for determining the location of the center of the hole for automated assembly.

Figure 6 shows the computed direction of the reaction moment vector for a square peg and hole when the peg z-rotation is close to the assembly orientation. Looking for patterns in the robot force/torque sensor signals to match these theoretical moments may provide a robot with the ability to infer the location of the assembly position and orientation as the assembly attempt proceeds.

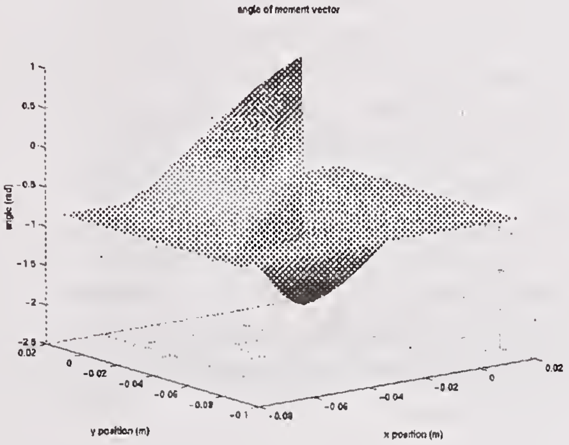


Fig. 6 Theoretical moment plot for a virtual square peg-in-hole simulation (peg z-rotation nearly aligned with hole)

CWRU researchers were able to show the dramatic effect that moments play in the human perception of assembly by feel. Experiments were conducted performing virtual-reality assemblies with a Cybernet force-reflecting hand controller. Mobility was restricted to the equivalent of a 4-dof SCARA robot (i.e., wrist pitch and roll were fixed). Computed interaction moments about the x and y axes (e.g., as shown in Fig 6) were reflected back to the hand controller as the operator attempted the virtual assembly. Without the benefit of vision, the operator had to search blindly in x and y (and z-rotation, for square pegs) for the insertion location. For a small-clearance assembly, a blind search without sensory feedback was typically unsuccessful within 2 minutes per trial. However, when the moment information was displayed haptically, human operators were able to learn how to interpret these signals and guide the peg to the assembly location. For round pegs, the mean assembly time for a trained operator using moment-feedback cues was less than 7 seconds.

While we can conclude that humans can exploit perception of reaction moments to dramatically improve assembly performance, it is not obvious how these signals are being interpreted and utilized. To help expose the essential features of such feedback cues, the moment information was deliberately corrupted before being presented to the user. Low-pass filtering, high-pass filtering and nonlinear transformations were performed to determine the influence on usability of the signals by humans. Figure 7 shows a record of high-pass filtered computed interaction moments presented to the operator during an assembly trial. This limited feedback was also successfully interpreted by humans to achieve rapid successful assemblies. This feedback was even further corrupted to display (exert) pulsed moments of fixed amplitude and duration when the rectified, high-pass filtered moment data exceeded a threshold. Remarkably, humans were just as adept at keying off of this impoverished feedback to achieve rapid assembly success. Such experiments are helping to identify the essential features of sensory feedback for expert, sensor-based assembly.

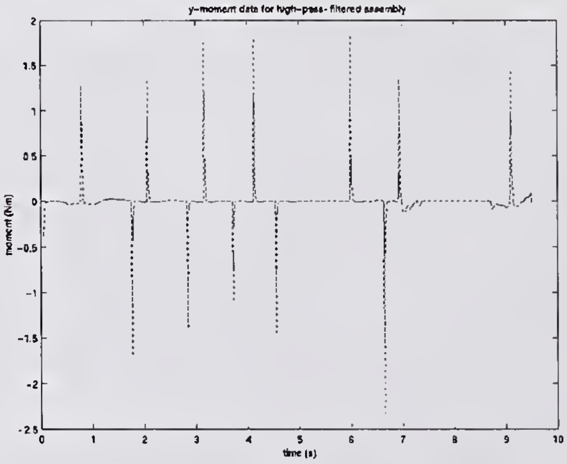


Fig. 7 Highpass filtered moment data is sufficient for humans to perform edge identification

In addition to testing which sensory cues are most effectively utilized by humans, we can also observe human assembly strategies from records of virtual assembly trials. Figures 8 and 9 show records of novice and trained human operators attempting the teleoperated virtual assembly of a round peg in a round hole. Both the experts and novices exhibit the behavior of attraction towards a sharp discontinuity in reaction moments (the vertical line segment separating large negative moments from large positive moments). The expert operator utilizes this discontinuity in a purposeful manner. The expert behavior shows an initial scan (apparently seeking the discontinuity boundary) followed by tracking the moment discontinuity boundary towards its apex. If the assembly location near the apex is overshot, the operator detects the loss of signal and loops back to restart the search within a small neighborhood of the solution.

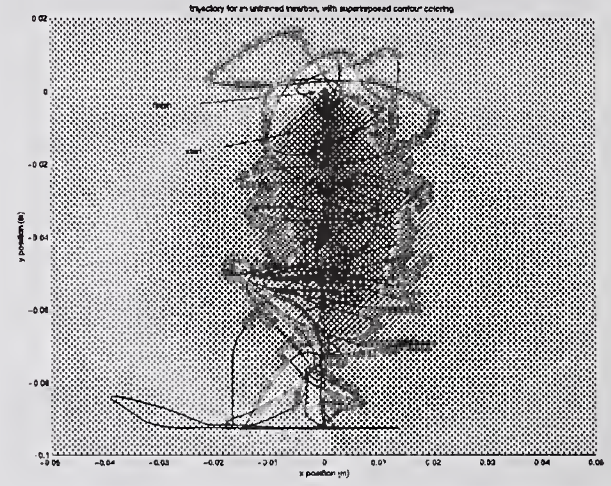


Fig.8 The path taken during an untrained assembly trial.

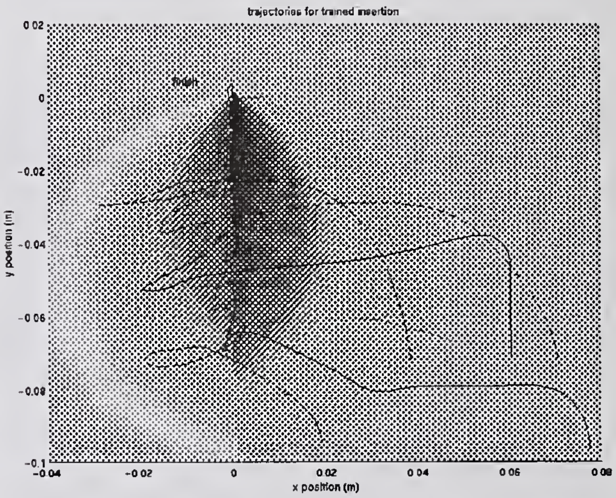


Fig. 9 The path taken during an trained assembly trial. Notice that this path is much more of an optimal trajectory to achieve assembly.

Based on such observations of human strategies for assembly, researchers at CWRU have been able to construct a neural-net based controller that exhibits similar sensory interpretation to perform guided assembly. Using an NAC-controlled AdeptOne robot equiped with a JRRR 6-axis force/torque sensor, a map of reaction moments vs displacement errors was experimentally obtained for a peg-in-hole assembly task. The acquired raw data was used to train a neural net to recognize assembly coordinates from x and y moment signals.

After training, the quality of the learned mapping was tested. Over a large region of displacements, the mapping was untrustworthy. In contrast, over a small region, the mapping was reasonably reliable and precise. Significant additional regions were capable of coarse but useful predictions. These results are illustrated in Fig 10. The sparse x's indicate regions of high-quality predictive capability, the triangles and squares are regions of coarse but useful mappings, and the regions of diamonds and circles correspond to poor predictive capability.

Following the strategy of human operators, our intelligent controller incorporates a sensory-driven behaviors. When in the low-quality mapping regions, the robot is controlled to perform a compliant raster search for the hole. When useful sensor information becomes available, the robot alters its search direction as indicated by the vectors in Fig 10. The coarse information is sufficient to guide the robot towards the region of high information, after which the robot can follow a reasonably precise path towards the goal.

The raster-search phase is like the approach phase of the human operator illustrated in Fig 9. In the region of the moment discontinuity, the operator (and the robot) are drawn towards the discontinuity boundary. Upon reaching the discontinuity boundary, both the human operator and the robot follow this narrow region of high-quality sensory information towards the goal.

The intelligent control algorithm was tested on an AdeptOne robot controlled by NAC. It demonstrated searching behavior like that of humans for a simple (large-clearance, round peg-in-hole) task. For tighter clearances, however, the algorithm was not sufficiently robust. Subsequently, sensory interpretation was based on sensory patterns (moments vs time) rather than point-by-point measurements. With this augmentation, it was much easier to recognize key features (peaks and discontinuities) from a sequence of measurements, as opposed to depending on fortuitous samples within the slim region of high-quality information. It seems likely that humans perform similar processing, interpreting moment feedback in terms of temporal patterns rather than processing of distinct, point-wise measurements. (This would be consistent with, but not deducible from the recorded trajectories of Fig 10).

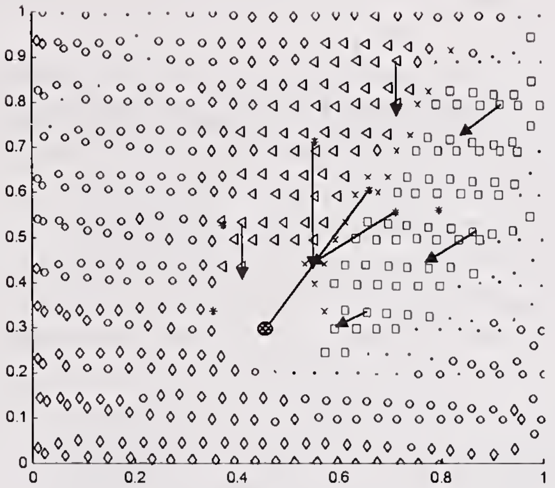


Fig. 10 NN calculated hole center direction vectors based on actual sensor feedback. Note: Where diamonds and circles are indicatedsensory cues are unreliable, whereas x's indicate high-information sensory data.

The above methods are promising for generating sensory-driven soft attractor trajectories. In addition to the goal directed manipulation of an attractor point, it is also possible to change the end-point dynamics of the robot by altering the spring and damper coefficients. The alterations of these parameters during the assembly process enable complex capabilities, such as tracking a trajectory quite stiffly (e.g. to approach a high-confidence target location) followed by a sudden relaxation of stiffness or damping parameters (e.g. to accommodate contact constraints while searching for an uncertain assembly location), as required by a particular application.

Such work in sensory-driven, behavior-based control may provide the foundation for realizing the third important quality of expert assembly exhibited by humans: the ability to improve performance through experience. It is our hope that our intelligent robot controller will be capable of autonomous data collection and autonomous neural-net training for automatic generation of programs for new assemblies. Such capability would eliminate the need for expensive, time-consuming robot programming and would enable robots to acquire expertise through experience

CONCLUSION

Attributes once thought only to belong exclusively to humans have now been demonstrated on the NIST FRAPA project using NAC robot controllers, vision, and neural networks. A large class of applications that have resisted automation due to positional uncertainty being greater than assembly tolerances, and the neeed to control the forces of contact of workpeices manipulated by robotics are now feasible.

In the near future robots will be commonly available that have the capacity to control their forces of contact, acquire and manipulate parts in uncertain locations and poses, and use trained neural networks to accomplish a predefined goal.

Acknowledgements:
We thank NIST for providing the FRAPA Joint Venture with the ATP award. Matt Collins of Perceptron for his work in 3-D vision. David Kozlowski from Sandia National Lab. The other members of the FRAPA Steering Committee: Connie Philips of the NCMS; Bob Stoughton of MicroDexterity Systems; Mike Yoshonis of PICO; Valerie Bolhouse, Frank Maslar, Steven Sparkes of Ford Motor Company. Contributors at CWRU are Ravi Hebbar and Dan Morris on low level controls, Yonghong Zhao, Craig Birkhimer and Prof. Yoh-Han Pao on force-guided strategies, and Prof. Michael Branicky, Siddharth Chhatpar, Yuesong Wang, and Ittichote Chuckpaiwong.

References

1. Robotic Industries Assoc. 1999

2. Whitney, D. 1987, "Historical Perspective and State of the Art in Robot Force Control," *International Journal of Robotics Research*, pp. 3-14, vol. 6, no.1.

3. Newman, W. S. and Mathewson, B.B., "Integration of Force Strategies and Natural admittance Control", Proceedings from 1994 International Mechanical Engineering Congress and Exposition, Vol. 1 of 2, Nov. 1994

4. Newman, W.S. and Y. Zhang, "Stable Interaction Control and Coulomb Friction Compensation Using Natural Admittance Control". *Journal of Robotic Systems*, 1994. **11**(1).

5. Newman, W.S., "Stability and Performance Limits of Interaction Controllers," *ASME J. Dynamic Systems, Measurement and Control*, 1992. 114(4): p. 563-570.

6. Pipitone, Frank and Adams, William, "Rapid Recognition of Freeform Objects in Noisy Range Images Using Tripod Operators", Navy Center for Applied Research and Artificial Intelligence (NCARAI), AIC-93-037, 1993

PART II
RESEARCH PAPERS

7. MULTIRESOLUTIONAL PHENOMENA IN INTELLIGENT SYSTEMS

Heterogeneous Computing

A. Wild
Motorola, Phoenix, AZ 85018

ABSTRACT

It is often considered, explicitly or implicitly, that constructed systems with autonomy must reflect, and potentially duplicate, the perceived capabilities of human intelligence, including having the ability to handle heterogeneity, e.g. to perform numerical computing, as well as various types of non-numerical computing. Since our knowledge is getting obsolete, no matter how much wisdom will be included in an intelligent system, it will loose its sharpness in time, unless it can improve itself. This would require changing system domains and internal interfaces, and selecting architectures that would support self-change.

KEYWORDS: *Heterogeneity, non-numerical computing, domain, interface, architecture.*

1. INTRODUCTION

Several authors pointed out capabilities that an intelligent system should possess, in addition to its ability to handle numbers. According to Merriam-Webster On-line Thesaurus, "compute" has as etymology the Latin "computare", from com- + putare, meaning "to consider", a much wider meaning that the contemporary usage of the verb "to compute", listed by the same source as being: 1 : to make calculation; 2 : to use a computer.

If anybody had any doubts, this historically wider perspective confirms that non-numerical computation is no oxymoron, but a legitimate area of research. However, when considering an intelligent system, it is easy to see that the question is not so much whether the right way to go is numerical or non-numerical computation : quite obviously, they should both be present among the system capabilities.

Furthermore, "non-numerical" is a simplification, reducing the number of cases to two, "tertium non datur". Actually, the generic "non-something" will spontaneously split into any number of "somethings". An intelligent system needs to be able to handle "all of the above", and more.

This paper lists some questions that have been partially addressed, from this perspective, and/or might be worth pursuing. It does not contain a corresponding list of answers, as most of them are expected from future research. It is rather initiating a whish list.

2. EVOLUTION

At any point in time, our understanding of the world is imperfect, as it is:

- incomplete, not being able to explain all we observe
- contradictory, containing different and mutually incompatible explanations of the same facts
- partially wrong, as many explanations currently accepted are likely to be falsified in the future.

We may try to incorporate all our knowledge at this point in time, or any part thereof we think appropriate, into a constructed system. But, sooner or later, unavoidably, it will become hopelessly outdated, unless it would have a built-in capability to progress. Obviously, this capability would be an important feature even at lower levels of resolution, and for simpler tasks, than maintaining an internal image of the world.

Of particular interest would be whether the system could evolve in synchronism with our understanding of the human mind, as the theories about the human mind provide a major source of inspiration for constructed systems. Their evolution would surely add new content to be implemented in constructed systems. But, in general terms, this would be the effect of any advancement, in any area of knowledge.

How can a constructed system evolve, without human intervention ? Can it include domains that became relevant after the system was built ? Can and should it modify or eliminate some of the domains implemented at its "birth", that turn out as being wrong or irrelevant ?

3. SELF-STRUCTURING

A particular aspect of the evolution is the capability to modify the interactions between the parts of the system. In an evolving system, it is to be expected that the information exchange between domains would have to be adjusted continuously, "on the fly".

As a particular example, when evolving, the system must be able to define new interfaces between its parts, be they old or new. The changes may be initiated in response to different needs: eliminate computational bottlenecks, add new capabilities, eliminate useless parts of the system, etc. Even if no domains are added or eliminated, the system may determine that a different structure would have desirable advantages, and would take advantage of it by re-defining its own partitioning, information flow etc.

Is there a way for a system to control interfaces among its own sub-systems, e.g. define new ones, eliminate or modify existing ones ? How can a system re-architect itself ?

4. ARCHITECTURE, HIERARCHY

For some time, scientists pursued the idea that everything can be reduced to simple axioms, from which we would derive the apparent complexity of our surroundings. Still useful in particular disciplines, this hope has been largely replaced by a view accepting complexity as a fundamental feature of the world.

Usually, we handle complexity by introducing hierarchy. At each level, simpler concepts can be used to describe observations, while rules and procedures are established for crossing the boundaries from one hierarchical level to the next one.

In heterogeneous computing systems, each domain is likely to have some hierarchy, but assembling domains presents difficulties even in relatively simple cases. For instance, the hierarchical design database of an integrated circuit would not result automatically from merging hierarchical sub-circuits. If the same objects are rooted at different levels in the sub-circuits, the assembly will have most objects present at most hierarchical levels. Connections between sub-circuits will cross hierarchy borders, if the inputs/outputs of sub-circuits are at different levels. Various views of the objects in the database, such as timing, physical construction, etc., would not be propagated automatically up the hierarchy levels. To make merging possible without diluting or destroying the hierarchy, the designers of the sub-circuits must follow common, rigid rules. Alternatively, the system should automatically re-structure domain hierarchy. Today, systems do that only for the trivial case of one single level (flat hierarchy, actually no hierarchy !).

The problem is obviously more complicated when the system includes heterogeneous parts. Human programmers, exploring ad-hoc possibilities for connecting different domains of numerical computing, introduce transition domains, implementing rules for connecting space, time and parameters, e.g. by using interpolation/extrapolation rules in space, running the local times in lock step, and coding equations for parameters.

In more general terms, the requirement for an intelligent system would be to connect domains of various types of non-numerical computation, both with each other, and with domains of numerical computation.

How could a system be architected such that heterogeneous elements, like different types of computation (reasoning ?), may coexist, interact and add value to each other ? What would the interfaces between its domains be looking like ?

5. ONE, TWO, MANY

An intuitive way to build a system capable of acting upon itself is to architect it as a two-part structure: the first part addresses the tasks at hand, while the second part is optimizing the first part, acting like a conscience, or an ego of the system. This architecture seems to be able to ensure capabilities like evolving the first part of the system, or re-structuring it.

Another principle, probably much easier to envision than to implement, could build upon the paradigm of de-centralization. Any domain, facing difficulties in solving a task, would be entitled to start a browser, searching for useful capabilities in other domains. If the answer seems positive, interfaces would be put in place to connect the discovered resource with the domain trying to solve the task. If the browser does not provide a useful answer, a "generate domain" function could be started to fill the gap.

Is a non-hierarchical, self-configuring, heterogeneous system at all possible ? If yes, are there any rules to follow, are there impossible situations to avoid, or, alternatively, anything goes, and the solutions will be selected by trial and error ? Can this happen across hierarchical boundaries without generating unbearable chaos ?

6. IDENTITY, DREAMS

Allowing every domain to take the initiative in changing the system seems risky (yet democracy mostly works !). Clearly, in a two-part architecture, one part remains untouched, and can assume the task to ensure the stability of the system. In a de-centralized system, there may still be a need to define some parts as "untouchable", and a boundary might be needed to separate them from the parts that can be changed.

For one thing, the decentralized process envisioned above is likely to accumulate, over time, numerous useless connections, unnecessary search results, and other by-products. This suggests that the system would develop a need for a cleaning procedure. The system would "go to sleep", while running procedures that would tide it up. While "sleeping", it would be going through a sequence of abnormal states, strange and seemingly useless, that could be metaphorically called "dreams". The control of the system could be provided by a relatively simple and unintelligent mechanism, forcing it to undergo circadian cycles.

This line of thinking, this model and this metaphor may seem excessively anthropomorphic. Nonetheless, there may be sufficient reasons to allow for it, among other representations, in a system truly capable to handle all types of heterogeneous computing.

A Hierarchical Framework for Constructing Intelligent Systems Metrics

Ronald R. Yager
Machine Intelligence Institute
Iona College
New Rochelle, NY 10801

ABSTRACT

The focus of this work is on the development of a tool to enable the construction of performance metrics for intelligent systems which allows for the expression of intelligence in terms of high level concepts while allowing for the evaluation in terms of more basic measurable attributes.

1. Introduction

The measurement of performance of intelligent systems is clearly a context dependent process. This type of evaluation strongly depends upon the purpose for which the system is being used and the types of "intelligence" it is required to manifest.. However independent of the context the construction of such performance metrics requires the ability represent sophisticated human concepts needed to describe the various aspects of intelligence. While the expression of what constitutes intelligent performance may involve high level cognitive concepts the actual calculation of performance must be based upon measurable attributes associated with the system. The focus of this work is on the development of a tool to enable the construction of performance metrics for intelligent systems which allows for the expression of intelligence in terms of high level concepts while allowing for the evaluation in terms of more basic measurable attributes. This framework, which makes considerable use of the Ordered Weighted Averaging (OWA) operator [1, 2], also supports a hierarchical structure which allows for an increased expressiveness.

2. A General Approach to Aggregation

Central to any tool used for construction of intelligent systems metrics is the need for the aggregation of scores. In order to provide a very general framework to implement aggregations, we shall use the Ordered Weighted Averaging (OWA) operator [1, 2]. In the following, we briefly review the basic ideas associated with this class of aggregation operators.

Definition: An Ordered Weighted Averaging (OWA) operator of dimension n is a mapping F which has an associated weighting vector W such that its components w_j satisfy the following conditions 1. $w_j \in [0,1]$ and 2. $\sum_{j=1}^{n} w_j = 1$ and where$F(a_1, a_2,..., a_n) = \sum_{j=1}^{n} w_j b_j$ with b_j being the j^{th} largest of the a_i.

A key feature of this operator is the ordering of the arguments by value, a process that introduces a nonlinearity into the operation. Formally, we can represent this aggregation operator in vector notation as $F(a_1, a_2,..., a_n) = W^T B$, where W is the weighting vector and B is a vector, called the ordered argument vector, whose components are the b_j. Here we see the nonlinearity is restricted to the process of generating B. It can be shown that this operator is in the class of mean operators as it is commutative, monotonic, and bounded, $\text{Min}[a_i] \leq F(a_1, a_2,..., a_n) \leq \text{Max}[a_i]$. It can also be seen to be idempotent, $F(a, a,..., a) = a$.

The great generality of this operator lies in the fact that by selecting the w_j, we can implement many different aggregation operators. Specifically, by appropriately selecting the weights in W, we can emphasize different arguments based upon their position in the ordering. If we place most of the weights near the top of W, we can emphasize the higher scores, while placing the weights near bottom of W emphasizes the lower scores in the aggregation.

A number of special cases of these operators have been pointed out in the literature [3]. Each of these special cases is distinguished by the structure of the weighting vector W. Consider the situation where the weights are such that $w_1 = 1$ and $w_j = 0$ for all $j \neq 1$, this weighting vector is denoted as W^*. In this case we get $F(a_1, a_2,..., a_n) = \text{Max}_j[a_j]$. Thus the Max operator is a special case of the OWA operator. If the weights are such that $w_n = 1$ and $w_j = 0$ for $j \neq n$, denoted W_*, we get $F(a_1, a_2,..., a_n) = \text{Min}_j[a_j]$. Thus the Min operator is a special case of the OWA operator. If the weights are such that $w_j = \frac{1}{n}$ for all j, denoted W_{ave}, then $F(a_1, a_2,..., a_n) = \frac{1}{n} \sum_{j=1}^{n} a_j$ Thus we see that the simple average is also a special case of these operators.

If $W = W^{[k]}$ is such that $w_k = 1$ and $w_j = 0$ for $j \neq k$,then $F(a_1, a_2,..., a_n) = b_k$, the kth largest of the a_i. The median is also a special case of this family of operators. If n is odd, we obtain the median by selecting $w_{\frac{n+1}{2}} = 1$ and by letting $w_j = 0$, for $j \neq \frac{n+1}{2}$. If n is even, we get the median by selecting $w_{\frac{n}{2}} = w_{\frac{n}{2}+1} = \frac{1}{2}$ and letting $w_j = 0$ for all other terms.

An interesting class of these operators is the so-called olympic aggregators. The simplest example of this case is where we select $w_1 = w_n = 0$ and let $w_j = \frac{1}{n-2}$ for $j \neq 1$ or n. In this case, we have eliminated the highest and lowest scores and we've taken the average of the rest. We note that this process is often used in obtaining aggregated scores from judges in olympic events such as gymnastics and diving.

In [1], we introduced two measures useful for characterizing OWA operators. The first of these measures, called the alpha value of the weighting vector, is defined as $\alpha = \frac{1}{n-1} \sum_{j=1}^{n} (n - j)\, w_j$. It can be shown, $\alpha \in [0, 1]$. Furthermore, it can also be shown that if $W = W^*$ then $\alpha = 1$, if $W = W_{ave}$ then $\alpha = 0.5$ and if $W = W_*$ then $\alpha = 0$.

Essentially α provides some indication of the inclination of the OWA operators for giving more weight to the higher scores or lower scores. The closer α is to one, greater preference is given to the higher scores, the closer α is to zero, the greater preference is given to lower scores, and a value close to 0.5 indicates no preference. The actual semantics associated with α depends upon the application at hand. For example, in using the OWA operators to model logical connectives between the *and* and *or,* α can be associated with a measure of the degree of *orness* associated with an aggregation. .

It can be shown that while $\alpha = 1$ only if $W = W^*$ and $\alpha = 0$ only if $W = W_*$, other values of α can be obtained for many different cases of W. A particularly interesting case is $\alpha = 0.5$. It can be shown that for any OWA operator having a W with $w_{n-j+1} = w_j$ for all j, we get $\alpha = 0.5$. Thus we see any symmetric OWA operator has $\alpha = 0.5$. Essentially these operators are in the same spirit as the simple average.

The second measure introduced in [1] was

$$\text{Disp}(W) = -\frac{1}{n-1} w_j \ln(w_j).$$

In [1] it was suggested that this measure can be used to measure the degree to which we use all the information in the argument. It can be shown that for all W

$$0 \leq \text{Disp}(w) \leq \ln(n).$$

We note $\text{Disp}(w) = 0$ iff $W = W_{(k)}$ and $\text{Disp}(w) = \ln(n)$ iff $W = W_{ave}$. It can be shown that of all the symmetric implementations of W, those having $\alpha = 0.5$ (W_{ave}) has the largest measure of Disp.

3. Linguistic Description of OWA Operators

Let us now consider a basic application of the OWA operator. Assume $A_1, A_2,....., A_n$ is a collection of measurable attributes useful in characterizing intelligence in a system. For any given system d, let $A_i(d) \in [0, 1]$ indicate the degree it satisfies the property associated with attribute A_i. Using the OWA operator we can obtain a measure of satisfaction to this collection of attributes as $\text{Val}(d) = F_w(A_i(d), A_2(d), ..., A_n(d))$. Since the value obtained as a result of using the OWA aggregation is dependent upon the weighting vector, the issue of deciding upon the weighting vector appropriate for a particular aggregation is of great importance. One of the beneficial features of the OWA operator is the considerable number of different approaches that have been suggested for obtaining the weighting vector to use in any given application [4]. Of particular significance is the strong semantic underpinning of these approaches. This strong semantic connection allows users to easily translate their requirements, which may be expressed in many different ways, into appropriate OWA weighting vectors. Here we shall describe an approach based upon the idea of linguistic quantifiers.

The concept of linguistic quantifiers was originally introduced by Zadeh [5]. A linguistic quantifier, more specifically a proportional linguistic quantifier, is a term corresponding to a proportion of objects. While most formal systems, such as logic, allow just two quantifiers, *for all* and *there exists*, as noted by Zadeh, human discourse is replete with a vast array of terms, fuzzy and crisp, that are used to express information about proportions. Examples of this are *most*, *at least half*, *all*, *about* $\frac{1}{3}$. Motivated by this Zadeh [5] suggested a method for formally representing these linguistic quantifiers. Let *Q* be a linguistic expression corresponding to a quantifier such as *most*. Zadeh suggested representing this as a fuzzy subset Q over I = [0, 1] in which for any proportion $r \in I$, Q(r) indicates the degree to which r satisfies the concept indicated by the quantifier *Q*.

In [6] Yager showed how we can use a linguistic quantifier to obtain a weighting vector W associated with an OWA aggregation. For our purposes we shall restrict ourselves to regularly increasing monotonic (RIM) quantifiers. A fuzzy subset $Q : I \longrightarrow I$ is said to represent a RIM linguistic quantifier if: 1. Q(0) = 0, 2. Q(1) = 1 and 3. if $r_1 > r_2$ then $Q(r_1) \geq Q(r_2)$ (monotonic)

These RIM quantifiers model the class in which an increase in proportion results in an increase in compatibility to the linguistic expression being modeled. Examples of these types of quantifiers are *at least one, all, at least* α *%, most, more than a few, some.*

Assume Q is a RIM quantifier. Then we can associate with Q an OWA weighting vector W such that for j = 1 to n

$$w_j = Q(\frac{j}{n}) - Q(\frac{j-1}{n}).$$

Thus using this approach we obtain the weighting vector directly from the linguistic expression of the quantifier. The properties of RIMness guarantee that the properties of W are satisfied.

Let us look at the situation for some prototypical quantifiers. The quantifier *for all* is shown in figure #1. In this case we get that $w_j = 0$ for $j \neq n$, and $w_n = 1$, $W = W_*$ In this case we get as our aggregation the minimum of the aggregates. We also recall that the quantifier *for all* corresponds to the logical "anding" of all the arguments

In figure #2 we see the existential quantifier, *not none*. In this case $w_1 = 1$ and $w_j = 0$ for $j > 1$, $W = W^*$. This can be seen as inducing the maximum aggregation. It is recalled this quantifier corresponds to a logical *oring* of the arguments

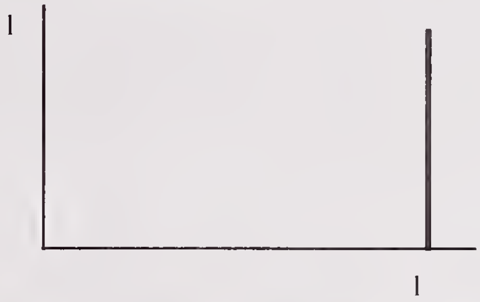


Figure #1. Linguistic quantifier "for all"

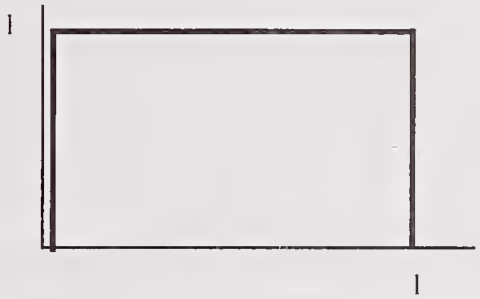


Figure #2. Linguistic quantifier "not none"

Figure #3 is seen as corresponding to the quantifier *at least* α. For this quantifier $w_j = 1$ for j such that $\frac{j-1}{n} < \alpha \leq \frac{j}{n}$ and $w_j = 0$ for all other

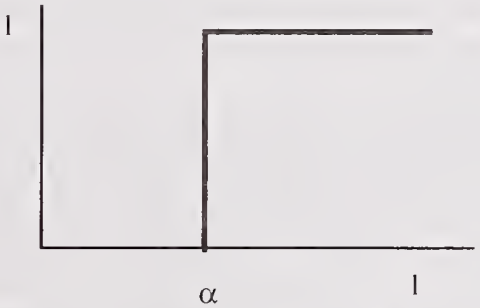


Figure #3. Linguistic quantifier "at least α"

Another quantifier is one in which Q(r) = r. Here we get $w_j = \frac{j}{n} - \frac{j-1}{n} = \frac{1}{n}$ for all j. This gives us the simple average. We denote this quantifier as *some.*

As discussed by Yager [3] one can consider parameterized families of quantifiers. Consider the parameterized family $Q(r) = r^\rho$, where $\rho \in [0, \infty]$. If $\rho = 0$, we get the existential quantifier; if $\rho = \infty$, we get *for all* and when $\rho = 1$, we are get the quantifier *some*. In addition for the case in which $\rho = 2$, $Q(r) = r^2$, we get one possible interpretation of the quantifier *most.*

As a result of the ideas so far presented here we can introduce the idea of a basic intelligence measuring module (IMM): $<A_1, A_2,.... A_n: Q>$, consisting of a collection of attributes and a linguistic quantifier indicating the proportion of the attributes we desire. Implicit in this module is the fact that the linguistic expression Q is essentially defining a weighting vector W for an OWA aggregation.

4. Including Attribute Importance

In the preceding we have indicated an IMM as consisting of a collection of attributes of interest and a quantifier Q indicating a mode of interaction between the attributes. Implicit in the preceding is the equal treatment of all attributes. For the construction of some intelligent systems measures we may need to ascribe differing importances to the attributes [4, 6]. In the following we shall consider the introduction of importance weights into our procedure.

Let $\alpha_i \in [0, 1]$ indicate the importance associated with attribute A_i. We assume $\alpha_i = 0$ indicates zero importance. With the introduction of these weights we can now consider a more general metric:

$$<A_1, A_2,...., A_n: M: Q>.$$

Here as before, the A_i are a collection of attributes and Q is a linguistic quantifier, however, here M is an n vector whose component $m_j = \alpha_j$, is the importance associated with A_j.

Our goal now is to calculate the overall score of a system d as $Val(d) = F_{Q/M}(A_1(d), A_2(d),...., A_n(d))$. Here $F_{Q/M}$ indicates an OWA operator. Our agenda here will be to first find an associated OWA weighting vector, W(d), based upon both Q and M. Once having obtained this vector we calculate Val(d) by the usual OWA process) = $W(d)^T B(d) = \sum_{j=1}^{n} w_j(d)\, b_j(d)$. Here $b_j(d)$ is the jth largest of the $A_i(d)$ and $w_j(d)$ is the jth component of the associated OWA vector W(d)

What is important to point out here is that, as we shall subsequently see, the weighting vector will be influenced by the ordering of the $A_i(d)$.

We now describe the procedure [4, 6] that shall be used to calculate the weighting vector, $w_j(d)$. The first step is to calculate the ordered argument vector B(d) such that $b_j(d)$ is the jth largest of the $A_i(d)$. Furthermore, we shall let μ_j denote the importance weight associated with the attribute that has the jth largest value. Thus if $A_5(d)$ is the largest of the $A_i(d)$, then $b_1(d) = A_5(d)$ and $u_1 = \alpha_5$. Our next step is to calculate the OWA weighting vector W(d). We obtain the associated weights as $w_j(d) = Q(\frac{S_j}{T}) - Q(\frac{S_{j-1}}{T})$ where $S_j = \sum_{k=1}^{j} u_k$ and $T = S_n$. T is the sum of all the importances and S_j is the sum of the importances of the jth most satisfied attributes. The following example will illustrate the use of this technique.

Example: Assume there are four attributes: A_1, A_2, A_3, A_4. The importances associated with these criteria are $u_1 = 1$, $u_2 = 0.6$, $u_3 = 0.5$ and $u_4 = 0.9$, giving us $T = 3$. We shall assume the quantifier guiding this aggregation is *most,* which is defined by $Q(r) = r^2$. Assume we have two system we are comparing, x and y, and the satisfactions to each of the attributes are:

$A_1(x) = 0.7$, $A_2(x) = 1$, $A_3(x) = 0.5$ and $A_4(x) = 0.6$

$A_1(y) = 0.6$, $A_2(y) = 0.3$, $A_3(y) = 0.$ and $A_4(y) = 1$

We first consider the valuation for x. In this case the ordering of the criteria satisfactions gives us:

	b_j	u_j
A_2	1	0.6
A_1	0.7	1
A_4	0.6	0.9
A_3	0.5	0.5

Calculating the weights associated with x, we get: $w_1(x) = 0.04$, $w_2(x) = 0.24$, $w_3(x) = 0.41$ and $w_4(x) = 0.31$. Using this $Val(x) = \sum_{j=1}^{4} w_j(x)\, b_j = 0.609$.

To calculate the score for y we proceed as follows. In this case the ordering of the criteria satisfaction is

	b_j	u_j
A_4	1	0.9
A_3	0.9	0.5
A_1	0.6	1
A_2	0.3	0.6

The associated weights are $w_1(y) = 0.09$, $w_2(y) = 0.13$, $w_3(y) = 0.42$and $w_4(y) = 0.36$ we then calculate

Val(y) = $\sum_{j=1}^{4} w_j(y)\ bj = 0.567$. Using this metric we see that system x would be deemed more intelligent.

More details with respect to the properties of this methodology can be found in [4, 6], however here we shall point out some properties associated with this approach. It can be shown that any attribute that has importance weight zero no affect on the result.

Consider the situation when all the attributes have the same importance, $\alpha_j = \alpha$. In this case $w_j(d) = Q(\frac{1}{n\,\alpha}\sum_{k=1}^{j} \alpha) - Q(\frac{1}{n\,\alpha}\sum_{k=1}^{j-1} \alpha) = Q(\frac{j}{n}) - Q(\frac{j-1}{n})$. This is the same set of weights we obtained when we didn't include any information with respect to importance.

Let us now look at the form of aggregation function obtained for some special cases of linguistic quantifiers. In the following we shall assume, without loss of generality, that the indexing is such that $A_i(d) \geq A_j(d)$ if $i < j$. Furthermore we shall suppress the d and denote $A_i(d) = a_i$. Using this notational convention

$$\text{Val}(d) = F_{Q/\alpha}(a_1, a_2, \ldots. a_n) = \sum_{j=1}^{n} a_j\, w_j$$

where $w_j = Q(\frac{1}{T}\sum_{k=1}^{j} \alpha_k) - Q(\frac{1}{T}\sum_{k=1}^{j-1} \alpha_k)$

Consider first the quantifier *some* , $Q(r) = r$. For this quantifier $w_j = \frac{\alpha_j}{T}$ and hence $\text{Val}(d) = \frac{1}{T}\sum_{j=1}^{n} \alpha_j\, a_j$

This is simply the weighted average of the attributes.

Consider now the case of the quantifier *for all*, $Q(1) = 1$ and $Q(r) = 0$ for $r \neq 1$. In this case $w_j = 0$ unless $\sum_{k=1}^{j} \alpha_k = T$ and $\sum_{k=1}^{j-1} \alpha_k < T$. We see $w_j = 1$ for the attribute having the smallest satisfaction and non-zero importance, hence $\text{Val}(d) = \underset{\alpha_j \neq 0}{\text{Min}}\ [a_j]$. For the case of the *existential* quantifier, $Q(0) = 0$ and $Q(r) = 1$ for all $r \neq 0$, we can easily show that $\text{Val}(d) = \underset{\alpha_j \neq 0}{\text{Max}}\ [a_j]$

Another example of a quantifier is the median quantifier. Here $Q(r) = 0$ for $r < 0.5$ and $Q(r) = 1$ for $r \geq 0.5$. In this case it can be shown that Val(d) can be obtained by the following simple process. First we normalize the weights, $\hat{\alpha}_j = \frac{\alpha_j}{T}$. Next we order the attribute scores in descending order and associate with each its normalized weight. We then, starting from the top, the highest score, add the normalized weights until we first reach a total of 0.5, the score of that attribute at which this total is reached is the aggregated value.

An interesting example of an OWA aggregation is the olympic aggregation. Here $w_1 = w_n = 0$ and $w_j = \frac{1}{n-2}$ for $j \neq 1$ or n. Using this aggregation we eliminate the highest and lowest scores and then take the average of the remaining scores. We can provide a generalization of this type of aggregation using a quantifier shown in figure #4.

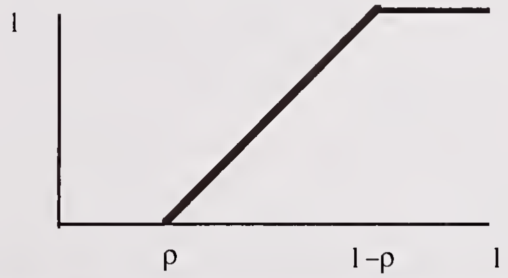


Figure #4. Generalized Olympic Quantifier

For this case

$$Q(r) = 0 \quad r < \rho$$
$$Q(r) = \frac{r - \rho}{1 - 2\rho} \qquad \rho \leq r \leq 1 - \rho$$
$$Q(r) = 1 \quad r > 1 - \rho$$

Here $w_j = 0$ for all j for which $\sum_{k=1}^{j} \frac{\alpha_k}{T} < \rho$ Similarly, $w_j = 0$ for all j for which $\sum_{k=j}^{n} \frac{\alpha_k}{T} > 1 - \rho$. In the range in between $w_j = \frac{\alpha_j}{1 - 2\rho}$

Another interesting example of OWA aggregation, one that is in some sense a dual of the olympic aggregation, is the so called Arrow-Hurwicz aggregation [7]. Here $w_1 = \alpha$ and $w_n = 1 - \alpha$, and $w_j = 0$ for all other. In this case we just consider the extreme values and eliminate the middle values. We can provide a generalization of this type of aggregation, one that can be used with importance weighted attributes, by introducing the quantifier shown in figure#5.

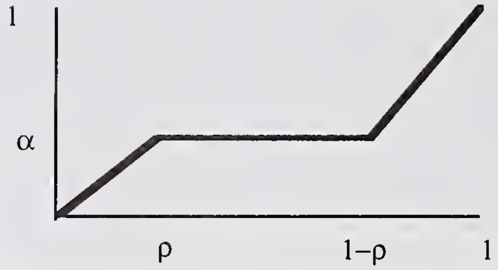


Figure #5. Generalized Arrow-Hurwicz

For this quantifier: $Q(r) = \frac{\alpha}{\rho}\ r$ if $r < \rho$, $Q(r) = \alpha$ if $\rho \leq r < 1 - \rho$ and $Q(r) = 1 - \frac{1 - \alpha}{\rho}\ (1 - r)$ if $r \geq 1 - \rho$. It is assumed $\rho \leq 0.5$. For this quantifier the weights used in the OWA aggregation are such that for the highest scoring attributes, those accounting for ρ portion of the importance, $w_j = \frac{\alpha}{\rho}$, for the least satisfied attributes, those accounting for ρ portion of the importance, $w_j = \frac{1 - \alpha}{\rho}$ and the middle scoring attributes $w_j = 0$. In this quantifier α can be seen as a degree of optimism and 1 - ρ as an indication of the extremism of the aggregation. A number special cases of this quantifier are worth noting. If $\rho = 0$ then we have $w_1 = \alpha$ and $w_n = 1 - \alpha$, the basic Arrow-Hurwicz aggregation. If $\alpha = \rho = 0.5$ then we get the quantifier $Q(r) = r$. If $\alpha = 1$ then we get the quantifier *at least ρ* and if $\alpha = 0$ then we get the quantifier *at least 1 − ρ.*

5. Including Priorities

In the preceding we have described a method for measuring the intelligence of a system based upon the metric $<A_1, A_2, ..., A_n: M: Q>$where the component α_j of the vector M indicates the weight associated with the attribute A_j. Implicit in our formulation was the idea that the weight α_j was explicitly provided by the user. This is not necessarily required. It is possible for the weight associated with attribute A_j to be determined by some property of the system itself. Thus let B_j be some measurable attribute associated with a system, and let $B_j(d)$ be the degree to which d satisfies this attribute. Then without introducing any additional complexity we can allow $\alpha_j(d) = B_j(d)$. Thus here the weight associated with attribute A_j depends the value $B_j(d)$. Thus within this framework we have the option of specifying the importance weights conditionally or non-conditionally or not at all.

Typically the association of importance weights with attributes reflects some measure of trade-off between the worth of the attributes. For example, consider the averaging operator where $\text{Val}(d) = \sum_{j=1}^{n} A_j(d)\ \alpha_j$. We see that a gain of Δ in $A_j(d)$ results in an increase in overall evaluation of $\alpha_j\ \Delta$, while a gain of Δ in $A_i(d)$ is worth an increase of $\alpha_i\ \Delta$. In particular, if $\alpha_j = 2$ and $\alpha_i = 1$, then we are willing to trade a gain of Δ in A_j for a loss of less than 2Δ in A_i.

In some cases where we desire two attributes, we may not be willing to trade-off one of for the other. For example, in evaluating the performance of an "intelligent" car, while we would like both safety and mileage efficiency, we are not willing to give up safety for efficiency. Such a situation implies the existence of a **priority** between the attributes, safety has priority over cost. In the following we suggest a mechanism for the inclusion of priority type relationships.

Assume A_1 and A_2 are two attributes for which there exists a priority relationship: A_1 has priority over A_2. In order to manifest this priority relationship, we require that the importance associated with A_2 be dependent upon the satisfaction of attribute A_1 by the system being evaluated. Here then $\alpha_2(d) = A_1(d)$. Let us investigate this idea for the simple weighted average. Assuming α_1 is fixed, we get

$$\text{Val}(d) = \alpha_1\ A_1(d) + A_1(d)\ A_2(d) = A_1(d)(\alpha_1 + A_2(d))$$

Here we see that if $A_1(d)$ is low, the contribution of $A_2(d)$ becomes small and hence it is not possible for a high value of A_2 to compensate for a low satisfaction to A_1. Thus if a system scores low on A_1 it will get a low rating.

More generally, consider the quantifier Q and assume A_i has priority over A_j. To implement this priority we make the importance associated with A_j related to the satisfaction of A_i. In particular, we let $\alpha_j = \alpha\ A_i$, where $\alpha \in [0, 1]$. Using this we get for the weight w_j associated with A_j that

$$w_j = Q(S_{k-1} + \frac{\alpha\ A_i(d)}{T}) - Q(S_{k-1})$$

where $S_{k-1} = \frac{\sum_{k=1}^{j-1} \alpha_k}{T}$. We see that as $A_i(d)$ gets smaller, the value w_j will decrease.

6. Concepts and Hierarchies

Throughout the preceding we have assumed a collection of attributes characterized by the fact that for any d we have available $A_i(d) \in [0,1]$, we say that the value of attribute A_j is directly accessible, we call A_j ground attribute. We shall now introduce a more general idea which we shall call an **I**ntelligence **M**easuring ***Concept* (*IM Concept*)**. We define an IM Concept as an object whose measure of satisfaction can be obtained for any system d. It is clear that the ground attributes are examples of concepts, they are special concepts in that their values are directly accessible.

Consider now the intelligence measuring module of the type we have previously introduced, it is of the form $<A_1, A_2, ..., A_q: M: Q>$. As we have indicated the evaluation of this for any system d can be obtained using our aggregation process. In the light of this observation, we can consider this object to be a IM concept, with Con = $<A_1, A_2, ..., A_q: M: Q>$ then its evaluation is $Con(d) = F_{Q/M}(A_1(d), A_2(d), ..., A_q(d))$. A special concept is an individual attribute, Con = $<A_j : M: Q> = A_j$,we shall call these atomic concepts. These atomic concepts require no Q or M, but just need an A_j specification.

The basic componentsat these IM Concepts are the attributes, the A_j. However, from a formal point of view, the ability to evaluate these type of concepts is based upon the fact that for any d we have a value $A_j(d)$. As we have just indicated a concept also has this property, for any d we can obtain a measure of its satisfaction. This observation allows us to extend our idea of IM concepts to allow for IM concepts whose evaluation depends upon other concepts. Thus we can consider IM concepts of the form

$$Con = <Con_1, Con_2, ..., Con_n: M: Q>.$$

Here each of the Con_j are concepts used to determine the satisfaction of Con by an aggregation process where M determines the weight of each of the participating concepts and Q is the quantifier guiding the aggregation of the component concepts.

The introduction of concepts into the intelligence measuring results in a hierarchical structure for the construction of metrics. Essentially, we unfold until we end up with atomic attributes which we can directly evaluate. The following simple examples illustrate the structure.

Example: Consider here the measure

$$(A_1 \text{ and } A_2 \text{ and } A_3) \text{ or } (A_3 \text{ and } A_4).$$

We consider this as a concept $<Con_1, Con_2 : M: Q>$.

Here Q is the existential quantifier and $M = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ In addition

$$Con_1 = <A_1, A_2, A_3: M_1: Q_1>$$
$$Con_2 = <A_3, A_4 : M_2: Q_2>$$

Here $Q_1 = Q_2 = \textit{all}$ and $M_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ and $M_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$.

This can be expressed in a hierarchical fashion, see figure #6.

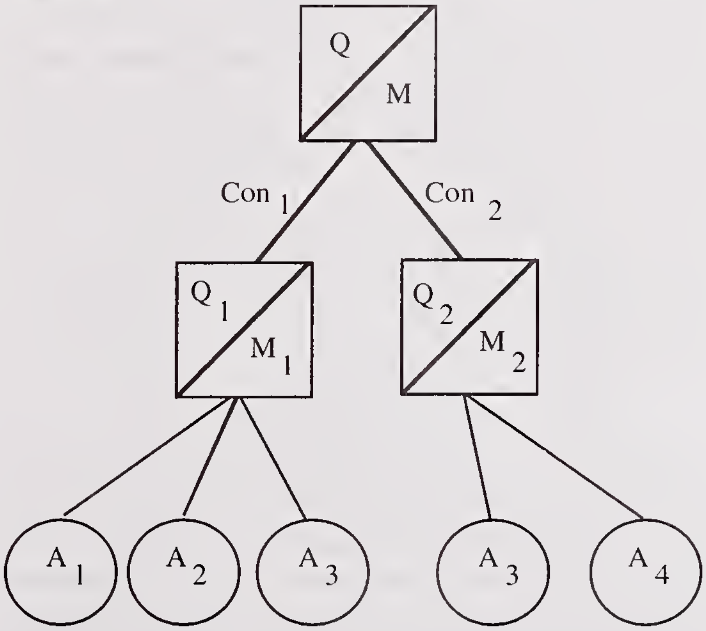


Figure #6. Hierarchical Formulation

An often used construct is the logical **if ... then** specification expressing the desire for some attribute if some other attribute is present. In a the following we describe a method for modeling this type of structure within our our framework.

Consider (A_1 **and** A_2) **or** (**if** A_3 **then** A_4). Figure #7 provides its hierarchical expansion.

In constructing this hierarchical implementation, we used the fact that “if A_3 then A_4” is logically equivalent to “not(A_3) or A_4.” Thus in this framework we interpret the concept "**if** A **then** B" as the concept $\overline{A}$ **or** B. We note that $\overline{A}(d) = 1 - A(d)$. More generally, the expression

if A_1 **and** A_2 **and** A_3 **then** B

is seen as equivalent to the expression $\overline{A}_1$ **or** $\overline{A}_2$ **or** $\overline{A}_3$ **or** B. This is represented as concept of the form <$\overline{A}_1$ $\overline{A}_2$, $\overline{A}_3$, B: -:*Or*>. We note the importances have not been specified and hence by default are all assumed to be one.

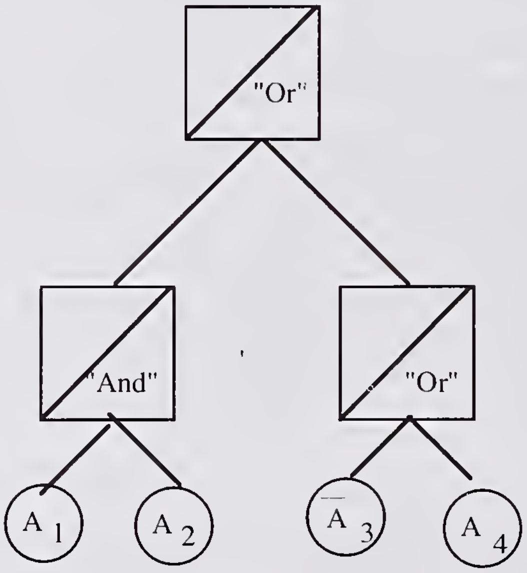


Figure #7. Implementation of if ... then

Using the ideas presented in the preceding we have a tool that can be used to construct complex measures of intelligent system performance. Using this tool we start with a high level expression of the appropriate measure. We then decompose this expression into the aggregation simpler concepts. We proceed in this manner until we obtain a characterization of our desired measure in terms of ground attributes which can be directly measured.

7. References

[1]. Yager, R. R., "On ordered weighted averaging aggregation operators in multi-criteria decision making," IEEE Transactions on Systems, Man and Cybernetics 18, 183-190, 1988.

[2]. Yager, R. R. and Kacprzyk, J., The Ordered Weighted Averaging Operators: Theory and Applications, Kluwer: Norwell, MA, 1997.

[3]. Yager, R. R., "Families of OWA operators," Fuzzy Sets and Systems 59, 125-148, 1993.

[4]. Yager, R. R., "On the inclusion of importances in OWA aggregations," in The Ordered Weighted Averaging Operators: Theory and Applications, edited by Yager, R. R. and Kacprzyk, J., Kluwer Academic Publishers: Norwell, MA, 41-59, 1997.

[5]. Zadeh, L. A., "A computational approach to fuzzy quantifiers in natural languages," Computing and Mathematics with Applications 9, 149-184, 1983.

[6]. Yager, R. R., "Quantifier guided aggregation using OWA operators," International Journal of Intelligent Systems 11, 49-73, 1996.

[7]. Arrow, K. J. and Hurwicz, L., "An optimality criterion for decision making under ignorance," in Uncertainty and Expectations in Economics, edited by Carter, C. F. and Ford, J. L., Kelley: New Jersey, 1972.

EXPLORATORY ANALYSIS ENABLED BY MULTIRESOLULTION, MULTIPERSPECTIVE MODELING

Paul K. Davis

RAND and the RAND Graduate School
1700 Main St., Santa Monica, CA 90407-2138, U.S.A.

ABSTRACT

The objective of exploratory analysis is to gain a broad understanding of a problem domain before going into details for particular cases. Its focus is understanding comprehensively the consequences of uncertainty, which requires a good deal more than normal sensitivity analysis. Such analysis is facilitated by multiresolution, multiperspective modeling (MRMPM) structures that are becoming increasingly practical. A knowledge of related design principles can help build interfaces to more normal legacy models, which can also be used for exploration.

1 BACKGROUND

Strategy problems are typically characterized by enormous uncertainties that should be central in assessment of alternative courses of action—although individuals and organizations often suppress those uncertainties and give a bizarre level of credence to wishful-thinking planning factors and other simplifications (Davis, 1994; Ch. 4; Davis, Gompert, and Kugler, 1996). In the past, an excuse for downplaying uncertainty analysis—except for marginal sensitivity analysis around some "best-estimate" baseline of dubious validity—was the sheer difficulty of doing better. The time required for setup, run, and analysis made extensive uncertainty work infeasible. Today, technology permits extensive uncertainty analysis with personal computers.

A key to treating uncertainty well is *exploratory analysis* (Davis and Hillestad, 2001). The objectives of exploratory analysis include understanding the implications of uncertainty for the problem at hand and informing the choice of strategy and subsequent modifications. In particular, *exploratory analysis can help identify strategies that are flexible, adaptive, and robust.* In successive sections, this paper describes exploratory analysis; puts it in context; discusses enabling technology and theory; points to companion papers applying the ideas; and concludes with some technology challenges for modeling and simulation. The paper draws heavily on a forthcoming book (Davis and Hillestad, 2001) and builds on a much rougher preliminary presentation of the same material (Davis, 2000).

2 EXPLORATORY ANALYSIS

2.1 What Exploratory Analysis Is and Is Not

Exploratory analysis examines the consequences of uncertainty. It can be thought of as sensitivity analysis done right, but is so different from usual sensitivity analysis as to deserve a separate name. It is closely related to scenario space analysis (Davis, 1994, Ch. 4) and "exploratory modeling" (Bankes, 1993; Lempert, et al., 1996). It is particularly useful for gaining a broad understanding of a problem domain before dipping into details. That, in turn, can greatly assist in the development and choice of strategies. It can also enhance "capabilities-based planning" by clarifying *when*—i.e., in what circumstances and with what assumptions about all the other factors—a given capability such as an improved weapon system or enhanced command and control will likely be sufficient or effective (Davis, Gompert, and Kugler, 1996). This contrasts with establishing a base-case scenario, and an organizationally blessed model and data base, and then asking "How does the outcome change if I have more of this capability?"

2.2 Types of Uncertainty

Uncertainty comes in many forms and it is useful (National Research Council, 1997) to distinguish between input uncertainties (i.e., parametric uncertainties) and structural uncertainty. Input uncertainty relates to imprecise knowledge of the model's input values. Structural uncertainty relates to questions about the form of the model itself: Does it reflect all the variables on which the real-world phenomenon purportedly described by the model depends? Is the analytical form correct? Some uncertainties may be inherent because they represent stochastic processes. Some may relate to fuzziness or imprecision, while others reflect discord among experts. Some relate to knowledge about the values of well-defined parameters, whereas others refer to future values that as yet have no true values.

It is convenient to express the uncertainties parametrically. If unsure about the model's form, we can describe this also to some extent with parameters. For example, paramcters may control the relative size of quadratic and exponential terms in an otherwise linear model. Or a discrete parameter may be a switch choosing among distinct analytical forms. Some parameters may apply to the deterministic aspect of a model, others to a stochastic aspect. For example, a model might describe the rate at which Red and Blue suffer attrition in combat according to a simplistic Lanchester square law:

$$\frac{d\tilde{R}}{dt} = -\tilde{K}_b\tilde{B}(t) \quad \frac{d\tilde{B}}{dt} = -\tilde{K}_r\tilde{R}(t)$$

where the attrition cocfficients for Red and Blue have both deterministic and stochastic parts, each of which are subject to uncertainty, as in (illustrating for Blue only)

$$\tilde{K}_b(t) = K_{bo}[1 + c_b\tilde{N}_b(t;\mu,\sigma_b)].$$

Here the N term is a normal random variable with mean μ and standard deviation σ. It represents stochastic processes occurring within a particular simulated war, e.g., from one time period to the next. The means and standard deviations are ordinary deterministic parameters, as are the coefficients K_{bo}, K_{ro}, c_r, and c_b. These have constant values within a particular war, but at what value they are constant is uncertain.

So far the equations have represented input uncertainty. However, suppose there is controversy over using the linear, square, or some hybrid version of a Lanchester equation. We could represent this dispute as input, or parametric, uncertainty by modifying the equation to read

$$\frac{d\tilde{R}}{dt} = -\tilde{K}_b\tilde{B}^e(t)\tilde{R}^f(t) \quad \frac{d\tilde{B}}{dt} = -\tilde{K}_r\tilde{B}^g(t)\tilde{R}^h(t).$$

Now, by treating the exponents as uncertain parameters, we could explore both input and structural uncertainties in the model—at least to some extent. The fly in the ointment is that nature's combat equations are much more complex (if they exist), and we don't even know their form. Suppose, merely as an example, that combatant effectiveness decays exponentially as combatants grow weary. We could not explore the consequences of different decay times if we did not even recognize the phenomenon in the equation's form. In fact, we *often* do not know the true system model. Nonetheless, much can be accomplished by allowing for diverse effects parametrically.

2.3 Types of Exploratory Analysis

Exploratory analysis can be conducted in several ways (Davis and Hillestad, 2001). Although most of the methods have been used in the past (see especially Morgan and Henrion, 1992), thcy are still not appreciated and are often poorly understood.

Input exploration (or *parametric exploration*) involves conducting model runs across the space of cases defined by discrete values of the parameters within their plausible domains. It considers not just excursions taken one-at-a-time as in normal sensitivity analysis relative to some presumed base-case set of values, but rather all the cases corresponding to value combinations defined by an experimental design (or a smaller sample). The results of such runs, which may number from dozens to hundreds of thousands or more, can be explored interactively with modern displays. Within perhaps a half-hour, a good analyst doing such exploration can often gain numerous important insights that were previously buried. He can understand not just which variables "matter," but *when*. For example, he may find that the outcome of the analysis may be rather insensitive to a given parameter for the so-called base case of assumptions, but quite sensitive for other plausible assumptions. That is, he may identify in what cases the parameter is important. To do capabilities-based planning for complex systems, this can be distinctly nontrivial.

A complement to parametric exploration is "*probabilistic exploration*" in which uncertainty about the input paramcters is represented by distribution functions representing the totality of one's so-called objective and subjective knowledge. I sometimes use quotes around "probability" because the distributions are seldom true frequencies or rigorous Bayesian probabilities, but rather rough estimates or analytical conveniences.

Using analytical or Monte Carlo methods, the resulting distribution of outcomes can be calculated. This can quickly give a sense for whether uncertainty is particularly important. In contrast to displays of parametric exploration, the output of probabilistic exploration gives little visual weight to improbable cases in which various inputs all have unlikely values simultaneously. Probabilistic exploration can be very useful for a condensed net assessment. Note that this use of probability methods is different from using them to describe the consequences of a stochastic process within a given simulation run. Indeed, one should be cautious about using probabilistic exploration because one can readily confuse variation across an ensemble of possible cases (e.g., different runs of a war simulation) with variation within a single case (e.g., fluctuation from day to day within a single simulated war). Also, an unknown constant parameter for a given simulated war is no longer unknown once the simulation begins and simulation agents representing commanders should perhaps observe and act upon the correct values within a few simulated time periods. Despite these subtleties, probabilistic exploration can be quite helpful.

The preferred approach treats some uncertainties parametrically and others with uncertainty distributions. That is, it is *hybrid exploration.* It may be appropriate to parameterize a few key variables that are under one's own control (purchases, allocation of resources, and so on), while treating the uncertainty of other variables through uncertainty distributions. One may also want also to parameterize a few variables characterizing the future context in which strategy must operate (e.g., short warning time). There is no general procedure here; instead, the procedure should be tailored to the problem at hand. In any case, the result can be a comprehensible summary of how known classes of uncertainty affect the problem at hand.

Let me give a few examples of what exploratory analysis can look like. Figure 1 mimics a computer screen during a parametric exploration of what is required militarily to defend Kuwait against a future Iraqi invasion by interdicting the attacker's movement with aircraft and missiles (Davis and Carrillo, 1997). Each square denotes the outcome of a particular model case (i.e., a specific choice of all the input values). The model being used depends on 10 variables–those on the x, y, and z axes, and seven listed to the side (the z–axis variable is also listed there, redundantly). The outcome of a given simulation is represented by the color (or, in this paper, by the pattern) of a given square. Thus, a white square represents a good case in which the attacker penetrates only a few tens of kilometers before being halted. A black square represents a bad case in which the attacker penetrates deep into the region that contains critical oil facilities. The other patterns represent in-between cases. The number in each square gives the penetration distance in km.

To display results in this way for a sizable scenario space RAND has often used a program called Data View, developed at RAND in the mid 1990s by Stephen Bankes and James Gillogly. After running the thousands or hundreds of thousands of cases corresponding to an experimental design for parametric exploration, we explore the outcome space at the computer. We can choose interactively which of the parameters to vary along the x, y, and z axes of the display. The other parameters then have the values shown along the right. However, we can click on their values and change them interactively by selecting from the menu of values for cases that have been run.

As mentioned above, in about a half an hour of such interactive work, one can develop a strong sense of how outcomes vary with *combinations* of parameter values. This is much more than traditional sensitivity analysis. Moreover, one can search out and focus upon the "good" cases. Figure 1 is merely one schematic snapshot of the computer screen for choices of parameter values that show some successes. Most snapshots would be dominated by black squares because it is difficult to defend Kuwait against a large threat. Data View is not a commercial product, but RAND has made it available to government clients and some other organizations (e.g., allied military staffs).

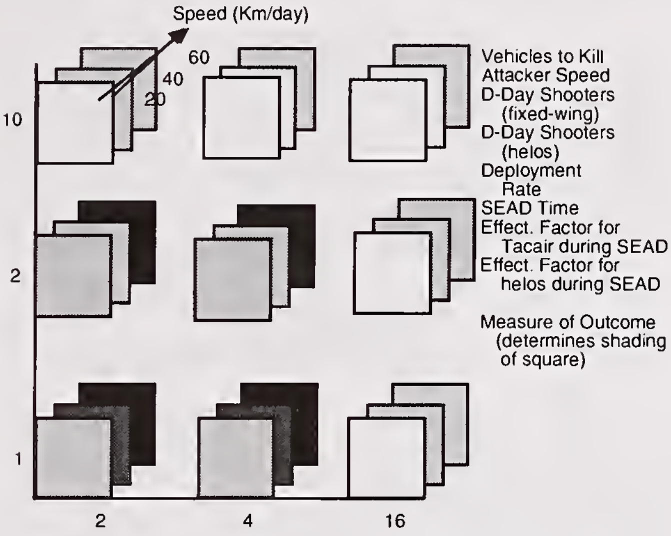
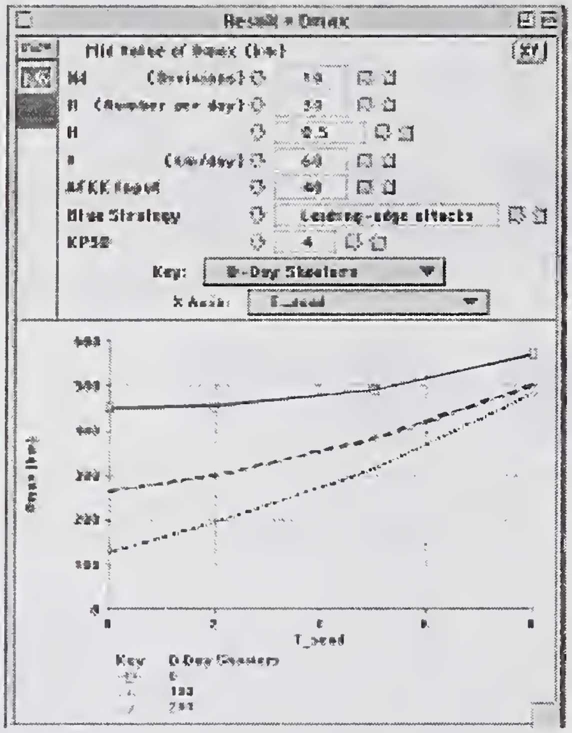


Figure 1: Display of Parametric Exploration

Other personal-computer tools can be used for the same purpose and the state of the art for such work is advancing rapidly. A much improved version of Data View called CAR™ is under development by Steve Bankes at Evolving Logic (`www.evolvinglogic.com`). For those who prefer spreadsheet modeling, there are plug-in programs for Microsoft EXCEL® that provide statistical capabilities and some means for exploratory analysis. Two of them are Crystal Ball® (`www.decisioneering.com`) and @Risk® (`www.palisade.com/html/risk.html`). For a number of reasons such as visual modeling and convenient array mathematics, I usually prefer the Analytica® modeling system (the exception is when one needs procedural programming). Analytica (`www.lumina.com`) is an outgrowth of the Demos system developed at Carnegie Mellon University (Morgan and Henrion, 1992).

Figure 2 shows a screen image from recent work with Analytica on the same problem treated in Figure 1. In this case, we have a more traditional graphical display. Outcome is measured along the Y axis and one of the independent variables is plotted along the X axis. A second variable (D-Day shooters) is reflected in the family of curves. The other independent variables appear in the rotation boxes at the top. As with Data View, we change parameter values by clicking on a value and selecting from a menu of values. Such interactive displays allow us to "fly through the outcome space" for many independent parameters, in this case 9. For this number, the display was still quickly interactive for the given model and computer (a Macintosh PowerBook G3 with 256 MB of RAM).

So far, the examples have focused on parametric exploration. Figure 3 illustrates a hybrid exploration (Davis, et al., 1998). It shows the distribution of simulation outcomes resulting from having varied most parameter values "probabilistically" across an ensemble of possible wars, but with warning time and the delay in attacking armored columns left parametric.



Result • Dmax

Key: D-Day Shooters

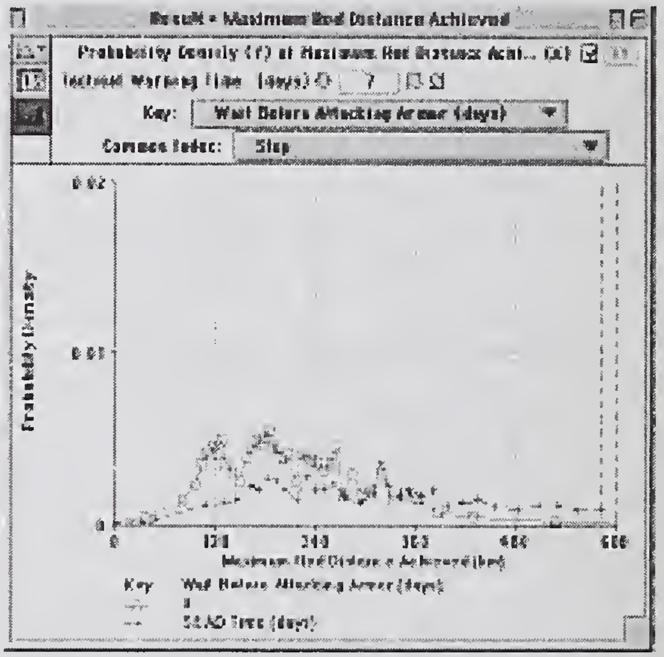
X Axis: T_dead

Figure 2: Analytica Display of Parametric Exploration

The probabilistic aspect of the calculation assumed, for example, that the enemy's movement rate had a triangular distribution across a particular range of values and that the suppression of air defenses would either be in the range of a few days or more like a week, depending on whether the enemy did or did not have air-defense systems and tactics that were not part of the best estimate. We represented this possibility with a discrete distribution for the likelihood of such surprises. The two curves in Figure 3 differ in that the one with crosses for markers assumes that interdiction of moving columns waits for suppression of air defenses (SEAD). The other curve assumes that interdiction begins immediately because the aircraft are assumed stealthy.

This depiction of the problem shows how widely the outcomes can vary and how the outcome distribution can be complex. The non-stealthy-aircraft case shows a spike at the right end where cases pile up because, in the simulation, the attacker halts at an objective of about 600km. Note that the mean is not a good metric: the "variance" is huge and the outcome may be multimodal.

These results have been from analyses accomplished in recent years for the Department of Defense. As we look to the future, much more is possible with computational tools. Much better displays are possible for the same information and, even more exciting, computational tools can be used to aid in the search process of exploration. For example, instead of clicking through the regions of the outcome space, tools could automatically find portions of the space in which particular outcomes are found. One could then fine-tune one's insights by clicking around in that much more limited region of the outcome space. Or, if the model is itself driven by the exploration apparatus, then the apparatus could search for outcomes of interest and then focus exploration on those regions of the input space. That is, the experimental design could be an output of the search rather than an input of the analysis process. These methods are at the core of the evolving tool mentioned earlier called CAR (for Computer-Assisted Reasoning).



Result • Maximum Red Distance Achieved

Key: Wait Before Attacking Armor (days)

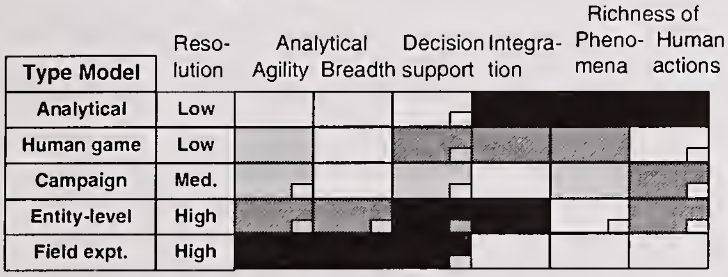
Common Index: Step

Figure 3: Analytica Display of "Probabilistic" Exploration

3 EXPLORATORY ANALYSIS IN CONTEXT

Exploratory analysis is an exciting development with a long history with RAND's RSAS and JICM models. How ever, it is only one part of a sound approach to analysis generally. It is worth pausing to emphasize this point. Figure 4 shows how different types of models and simulations (including human games) have distinct virtues. The figure is specialized to military applications, but a more generic version applies broadly to a wide class of analysis problems.

White rectangles indicate "good;" that is, if a cell of the matrix is white, then the type model indicated in the left column is very effective with respect to the attribute indicated in the cell's column. In particular, analytical models (top left corner), which have low resolution, can be especially powerful with respect to their analytical agility and breadth. In contrast, they are very poor (black cells) with respect to recognizing or dealing with the richness of underlying phenomena, or with the consequences of both human decisions and behavior. In contrast, field experiments often have very high resolution (they may be using the real equipment and people), and may be good or very good for revealing phenomena and reflecting human issues. They are, however, unwieldy and inappropriate for studying issues in breadth. The small insets in some of the cells indicate that the value of the type model for the particular purpose can often be enhanced a notch or two if the models include sensible decision algorithms or knowledge-based models that might be in the form of expert systems or artificial-intelligence agents.



Type Model	Resolution	Analytical Agility	Analytical Breadth	Decision support	Integration	Richness of Phenomena	Human actions
Analytical	Low	□	□	□	■	■	■
Human game	Low	░	□	▨	▨	▨	□
Campaign	Med.	░	□	░	□	░	▨
Entity-level	High	▨	▨	■	■	□	▨
Field expt.	High	■	■	■	□	□	□

Figure 4: Virtues of a Model and Gaming Family

Figure 4 was developed as part of an exhortation to the Department of Defense regarding the need to have *families of models* and *families of analysis* (Davis, Bigelow, and McEver, 1999). Unfortunately, government agencies often focus on a single model such as the venerable TACWAR, BRAWLER, or JANUS.

The niche of exploratory analysis is the top left hand corner of the matrix in Figure 4, which emphasizes analytical agility and *breadth* of analysis, rather than depth. However, the technique can be used hierarchically if one has a suitably modularized system model. One can do top-level exploration first and then zoom in. This is easier said than done, however, especially with traditional models. Specially designed models make things much easier, as discussed in what follows.

4 TECHNOLOGICAL ENABLERS

4.1 The Curse of Dimensionality

In principle, exploratory analysis can be accomplished with any model. In practice, it becomes difficult with large models. If F represents the model, it can be considered to be simply a complicated function of many variables. How can we run a computerized version of F to understand its character? If F has M inputs with uncertain values, then we could consider N values for each input, construct a full factorial design (or some subset, using an experimental design and sampling), run the cases, and thereby have a characterization. However, the number of such cases would grow rapidly (as N^M for full-factorial analysis), which quickly gets out of hand even with big computers. Quite aside from setup-and-run-time issues, comprehending and communicating the consequences becomes very difficult if M is large. Suppose someone asked "Under what conditions is F less than the danger point?" Given sufficiently powerful computers and enough time, we could create a data base of all the cases, after which we could respond to the question by spewing out lists of the cases in which F fell below the danger point. The list, however, might go on for thousands of pages. What would we do with the list? This is one manifestation of the curse of dimensionality.

4.2 The Need for Abstractions

It follows that, even if we have a perfect high-resolution model, we need abstractions to use it well. And, in the dominant case in which the high-resolution model is by no means perfect, we need abstractions that allow us to ponder the phenomena in meaningful ways, with relatively small numbers of cognitive chunks. People can reason with 3, 5, or 10 such cognitive chunks at a time, but not with hundreds. If the problem is truly complex, we must find ways to organize our reasoning. That is, we must decompose the problem by using principles of modularity and hierarchy. The need for an aspect of hierarchical organization is inescapable in most systems of interest—even though the system may be highly distributed and relatively nonhierarchical in an organizational sense.

A corollary of our need for abstractions is that *we need models that use the various abstractions as inputs*. It is not sufficient merely to display the abstracts as intermediate outputs (displays) of the ultimate detailed model. The reasons include the fact that when a decision maker asks a what-if question using abstractions, there is a 1:n mapping problem in translating his question into the inputs of a more detailed model. So also when one obtains macroscopic empirical information and tries to use it for calibration. Although analysts can trick the model by selecting a mapping, doing so can be cumbersome and treacherous. It is often better if the question can be answered by a model that accepts the abstractions as inputs.

4.3 Finding the Abstractions

Given the need for abstractions, how do we find them and how do we exploit them? Some guidelines are emerging (Davis and Bigelow, 1998).

4.3.1 When Conceiving New Models and Families

With new models, the issue is how to *design*. Several options here are as follows:

- Design the models and model families top down so that significant abstractions are built in from the start, but do so with enough understanding of the microscopics so that the top-down design is valid.

- Design the models and families bottom up, but with enough top-down insight to assure good intermediate-level abstractions from the start.
- Do either or both of the above, but with designs taken from different perspectives.

The list does not include a pure top-down or pure bottom-up design approach. Only seldom will either generate a good design of a complex system. Note also the idea of alternative perspectives. For example, those in combat arms may conceive military problems differently than logisticians, and even more differently than historians attempting a macro-view explanation of events.

4.3.2 When Dealing With Existing Models

Only sometimes do we have the opportunity to design from scratch. More typically, we must adapt existing models. Moreover, the model "families" we may have to work with are often families more on the basis of assertion than lineage. What do we then do? Some possibilities here are:

- Study the model and the questions that users ask of the model to discover useful abstractions. For example, inputs X, Y, and Z may enter the computations only as the product XYZ. Or a decision maker may ask questions in terms of concepts like force ratio. For mature models, the displays that have been added over time provide insights into useful abstractions.
- Apply statistical machinery to search for useful abstractions. For example, such machinery might test to see whether the system's behavior correlates not just with X,Y, and Z, but with XY, XZ, YZ, or XYZ.
- Idealize the system mathematically and combine this with physical insight or empirical observation to guess at the form of aggregate behavior (e.g., inverse dependence on one variable, or exponential dependence on another). Consider approximations such as an integral being the product of the effective width of the integration interval and a representative non-zero value of the integrand.

The first approach is perhaps a natural activity for a smart modeler and programmer who begins to study an existing program, but only if he open-minded about the usefulness of higher-level depictions. The second approach is an extension of normal statistical analysis. The third approach is a hybrid that I typically prefer to the second. It uses one's understanding of phenomenology, and theories of system behavior, to gain insights about the likely or possible abstractions *before* cranking statistical machinery.

4.3.3 The Problem with Occam's Razor

The principle of Occam's razor requires that we prefer the simplest explanation and, thus, the simplest model. Enthusiasts of statistical approaches tend to interpret this to mean that one should minimize the number of variables. They tend to focus on data and to avoid adding variables for "explanation" if the variables are not needed to predict the data. In contrast, subject-area phenomenologists may prefer to enrich the depiction by adding variables that provide a better picture of cause-effect chains, but go well beyond what can be supported with meager experimental data. My own predilection is that of the phenomenologist, but with MRM designs one can sometimes have one's cake and eat it: one can test results empirically by focusing on the abstract versions of a model, while using richer versions for deeper explanation.

As an aside, a version of the Occam's Razor principle emphasizes use of the explanation that is simplest enough to explain all there is to explain, but nothing simpler! This should include phenomena that one "knows about" even if they are not clearly visible in the limited data. I would add to this the admonition made decades ago by MIT's Jay Forrester that to omit showing a variable explicitly may be equivalent to assuming its value is unity.

Competition among approaches can be useful. For example, phenomenologists working a problem may be convinced that a problem must be described with complex computer programs having hundreds or thousands of data elements. A statistical analysis may show that, despite the model's apparent richness, the system's resulting behavior is driven by something much simpler. This, in turn, may lead to a reconceptualizing of the problem phenomenologically. Many analogues exist in physics and engineering.

4.3.4 Connections Between New and Old Models

Although the discussion in Section 4.3.2 distinguished sharply between the case of new models and old ones, the reader may have noticed connections. In essence, working with existing models should often involve sketching what the models *should* be like and how models with different resolution *should* connect substantively. That is, working with existing models may require us to go back to design issues. Individuals differ, but I, at least, often find it easier to engage the problem than to engage someone's else's idiosyncratically described solution. Furthermore, I then have a better understanding of assumptions and approximations.

With this background, let me now turn to the design of multiresolution, multiperspective models and families (Davis and Bigelow, 1999). Although this relates most directly to new models, it is relevant also to working with legacy models in preparing for exploratory analysis.

4.4 Multiresolution, Multiperspective Modeling

4.4.1 Definition

Multi-resolution modeling (MRM) is building a single model, a family of models, or both to describe the same phenomena at different levels of resolution, *and* to allow users to input parameters at those different levels depending on their needs. Variables at level n are abstractions of variables at level n+1. MRM is sometimes called variable- or selectable-resolution modeling. Figure 5 illustrates MRM schematically. It indicates that a higher level model (Model A) itself has more than one level of resolution. It can be used with either two or four inputs. However, in addition to its own MRM features, it has input variables that can either be specified directly or determined from the outputs of separate higher-resolution models (models B and C, shown as "on the side," for use when needed. In principle, one could attach models B and C in the software itself—creating a bigger model. However, in practice there are tradeoffs between doing that or keeping the more detailed models separate. For larger models and simulations, a combination single-model/family-of-models approach is desirable. This balances needs for analytical agility and complexity management.

MRM is not sufficient by itself because of the need for different abstractions or perspectives in different applications. That is, different perspectives—analogous to alternative representations in physics—are legitimate and important. They vary by conception of the system and choice of variables. Designing for both multiple resolution and multiple perspectives can be called MRMPM (pronounced Mr. MIPM).

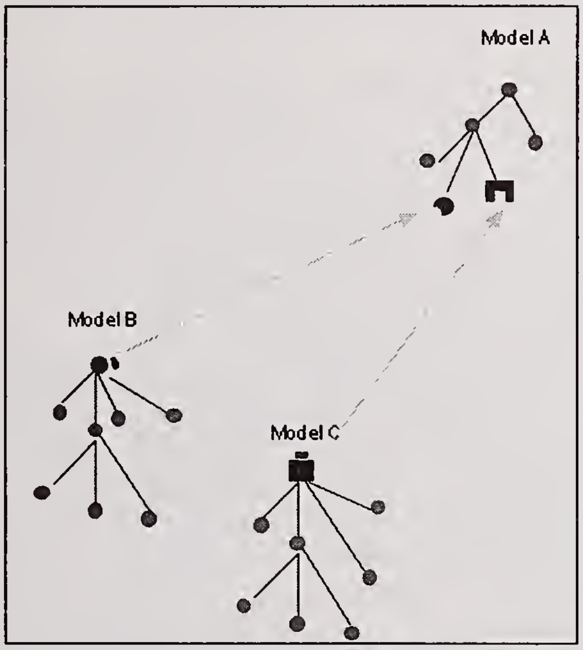


Figure 5: Figure 5: A Multiresolution Family

4.4.2 Mutual Calibration Within a Model Family

Given MRMPM models or families, we want to be able to reconcile the concepts and predictions among levels and perspectives. It is often assumed that the correct way to do this is to calibrate upward: treating the information of the most detailed model as correct and using it to calibrate the higher-level models. This is often appropriate, but the fact is that the more detailed models almost always have omissions and shortcomings. Further, different models of a family draw upon different sources of information—ranging from doctrine or even "lore" on one extreme to physical measurements on a test range at the other.

Figure 6 makes the point that members of a multiresolution model family should be *mutually* calibrated (National Research Council, 1997). For example, we may use low-resolution historical attrition or movement rates to help calibrate more detailed models predicting attrition and movement. This is not straightforward and is often done crudely by applying an overall scaling factor (fudge factor), rather than correcting the more atomic features of the detailed model, but it is likely familiar to readers. On the other hand, much calibration is indeed upward. For example, a combat model with attrition coefficients should typically have adjustments of those coefficients for different circumstances identified in a more detailed model.

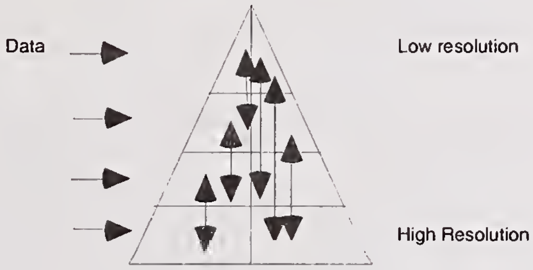


Figure 6: Mutual Calibration of Models in a Family

4.4.3 Design Considerations

So, given their desirability, how do we build a family of models? Or, given pre-existing models, how do we sketch out how they "should" relate before connecting them as software or using them for mutual calibration? Some highlights are as follows.

The first design principle is to recognize that there are limits to how well lower-resolution models can be consistent with high-resolution models. *Approximation is a central concept from the outset.* Several points are especially important:

- Consistency between two models should be assessed in the context of use. What matters is not whether they generate the same final state of the system, but whether they generate approximately the same results in the application (e.g., rank ordering of alternatives). This ties into the well-known concept of experimental frames (Zeigler, et al., 2000).
- Consistency of aggregated and disaggregated models must also be judged recognizing that low-resolution

models may reflect aggregate-level knowledge not contained in the detailed model.

- Comprehensive MRM is very difficult or impossible for complex M&S, but having even some MRM can be far more useful than having none at all.
- Members of an MRM family will typically be valid for only portions of the system's state space. Parameter values (and even functional forms) should change with region.
- Mechanisms are therefore needed to recognize different situations and shift models. In simulations, human intervention is one mechanism; agent-based modeling is another.
- Valid MRM will often require stochastic variables represented by probability distributions. Further, valid aggregate models must sometimes reflect correlations among variables that might naively be seen as probabilistically independent.

With these observations, the ideal for MRM is a hierarchical design for each MRM process, as indicated in Figure 5.

4.4.4 Desirable Design Attributes

From the considerations we have sketched above, it follows that models and analysis methodologies for exploratory analysis should have a number of characteristics. First, they should be able to reflect hierarchical decomposition through multiple levels of resolution and from alternative perspectives representing different "aspects" of a system.

Less obviously, they should also include realistic mechanisms for the natural entities of the system to act, react, adapt, mutate, and change. These mechanisms should reflect the relative "fitness" of the original and emerging entities for the environment in which they are operating. Many techniques are applicable here, including game-theoretic methods and others that may be relatively familiar to readers. However, the most fruitful new approaches are those typically associated with the term agent-based modeling. These include submodels that act "as the agents for" political leaders and military commanders or—at the other extreme— infantry privates on the battlefield or drivers of automobiles on the highway. In practice, such models need not be exotic: they may correspond to some relatively simple heuristic decision rules or to some well-known (though perhaps complex) operations-research algorithm. But to have such decision models is quite different from depending on scripts.

Because it is implausible that closed computer models will be able to meet the above challenge in the foreseeable future, the family of "models" should allow for human interaction—whether in human-only seminar games, small-scale model-supported human gaming, or distributed interactive simulation. This runs against the grain of much common practice.

4.4.5 Stochastic Inputs To Higher Level Models

The last item in the above list is often ignored in today's day-to-day work. Indeed, too often models that need to be stochastic are deterministic, with quantitatively serious consequences (Lucas, 2000). Often, workers calibrate a high-level (aggregate) model using average outcomes of allegedly "representative" high-resolution scenarios. For example, a theater-level model's air model might be calibrated to results of detailed air-to-air simulation with Brawler, which treats individual engagement classes (e.g., 1 on 1, 1 on 2, ... 4 on 8). This may appear to establish the validity of the theater-level model, but in fact the calibration is treacherous. After all, what kinds of engagements occur may be a sensitive function of the sides' command and control systems, strategies, and weather. The calibrations really nced to be accomplished on a highly study-specific basis.

Furthermore, the higher-level model inputs often need to be stochastic. Figure 7 illustrates the concept schematically for a simple problem. Suppose that a process (e.g., one computing the losses to aircraft in air-to-air encounters) depends on X,Y, S, and W. But suppose that the outcome of ultimate interest involves many instances of that process with different values of S and W (e.g., different per-engagement numbers of Red and Blue aircraft). An abstraction of the model might depend only on X,Y, and Z (e.g., overall attrition might depend on only numbers of Red and Blue aircraft, their relative quality, and some command and control factor). If the abstraction shown is to be valid, the variable Z should be consistent with the higher-resolution results. However, if it does not depend explicitly on S and W, then there are "hidden variables" in the problem and Z may appear to be a random variable, in which case so also would the predicted outcome F be a random variable. One could ignore this randomness if the distribution were narrow enough, but it might not be.

In the past, such calibrations have been rare because analysts have lacked both theory and tools for doing things better. The "theory" part includes not having good descriptions of how the detailed model should relate to the simplified one. The tool part includes the problem of being able to define the set of runs that should be done (representing the integral of Figure 7) and then actually making those runs.

Ideally, such a calibration would be dynamic within a simulation. Moreover, it would be easy to adjust the calibration to represent different assumptions about command, control, communications, computers, intelligence, surveillance, and reconnaissance (C4ISR), as well as tactics. We are nowhere near that happy situation today,

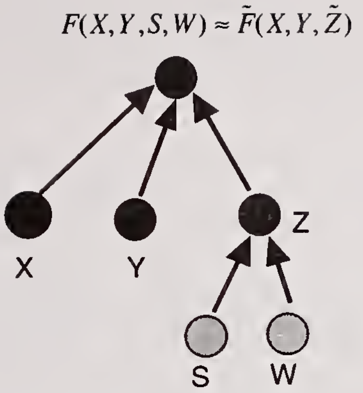


Figure 7: Input to Higher Level Model May Be Stochastic

5 RECENT EXPERIENCE AND CONCLUSIONS

MRMPM is not just idealized theory, but something usable. Over the last several years, my colleagues and I have done considerable work related to the problem of halting an invading army using precision fires from aircraft and missiles. The most recent aspects of that work included understanding in some detail how the effectiveness of such fires are affected by details of terrain, enemy maneuver tactics, certain aspects of command and control, and so on. This provided a good test bed for exploring numerous aspects of MRMPM theoiy (Davis, Bigelow, and McEver, 2000).

For this work we developed a multiresolution personal-computer model (PEM), written in Analytica, to understand and extend to other circumstances the findings from entity-level simulation of ground maneuver and long-range precision fires. A major part of that work was learning how to inform and calibrate PEM to the entity-level work. There was no possibility, in this instance, of revising the entity-level model. Nor, in practice, did we have such a good understanding of the model as to allow us to construct a comprehensive calibration theory. Instead, we had to construct a new, more abstract, model and attempt to impose some of its abstractions on the data from runs of the entity-level simulation in prior work, plus some special runs made for our purposes. The result is a case history with what are probably some generic lessons learned.

Figure 8 illustrates one aspect of PEM's design. It shows the data flow within a PEM module that generates the impact time (relative to the ideal impact time) for a salvo of precision weapons aimed at a packet of armored fighting vehicles observed by surveillance assets at an earlier time. Other parts of PEM combine information about packet location versus time and salvo effectiveness for targets that happen to be within the salvo's "footprint" at the time of impact, to estimate effectiveness of precision weapons. For the salvo-impact-time module, Figure 8 shows how PEM is designed to accept inputs as detailed as whether there is enroute retargeting of weapons, the latency time, and weapon flight time. However, it can also accept more aggregate inputs such as time from last update. If the input variable Resolution of Time of Last Update Calculation is set "low," then Time From Last Update is specified directly as input; if not, it is calculated from the lower-level inputs.

This design has proven very useful—both for analysis itself and for communicating insights to decision makers in different communities ranging from the C4ISR community to the programming and analysis community. In particular, the work clarified how the technology-intensive work of the C4ISR acquisition community relates to higher-level strategy problems and analysis of such problems at the theater level.

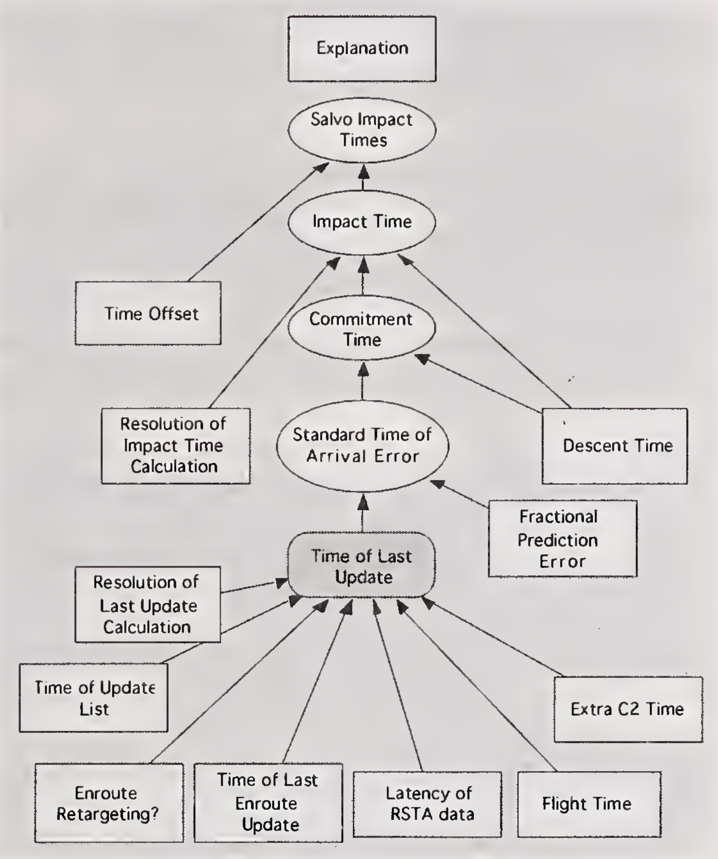


Figure 8: Multiresolution, Multiperspective Design

In other reports (McEver, Davis, and Bigelow, 2000a,b), we describe a broader but more abstract model (EXHALT) that we use for theater-level halt-problem analysis and experiments to deal with the multi-perspective problem. One conclusion is that MRMPM work rather demands a building-block approach that empasizes study-specific assembly of the precise model needed. Although we had some success in developing a closed MRMPM model with alternative user modes representing different demands for resolution and perspective (e.g., the switches in Figure 8), it proved impossible to do very much in that regard: the number of interesting user modes and resolution combinations simply precludes being able to wire in all the relevant user modes. Moreover, that explosion of complexity occurs very quickly. At-the-time-assembly from building blocks, not prior definition, is the stronger approach. This was as we expected, but even more so.

Fortunately, we were able to construct the models needed quickly—in hours rather than days or weeks—as the result of our building-block approach, visual modeling, use of array mathematics, and strong, modular, design.

We also concluded that current personal computer tools—as powerful as they are in comparison with those in past years—are not yet up to the challenge of making the building-block/assembly approach rigorous, understandable, controllable, and reproducible without unrealistically high levels of modeler/analyst discipline. Thus, there are good challenges ahead for the enabling–technology community. Also, the search models for advanced exploratory analysis are not yet well developed.

REFERENCES

Bankes, S. C. 1993. "Exploratory Modeling for Policy Analysis," *Operations Research*, Vol. 41, No. 3.

Bigelow, J. H., P. K. Davis, and J. McEver. 2000. "Case History of Using Entity-Level Simulation as Imperfect Experimental Data for Informing and Calibrating Simpler Analytical Models for Interdiction," *Proceedings of the SPIE*, Vol. 4026.

Davis, P. K (ed.). 1994. *New Challenges in Defense Planning: Rethinking How Much Is Enough*, RAND, Santa Monica, CA.

Davis, P.K.. 2000. "Multiresolution, Multiperspective Modeling as an Enabler of Exploratory Analysis," *Proceedings of the SPIE*, Vol. 4026.

Davis, P. K. and R. Hillestad. 2001. *Exploratory Analysis for Strategy Problems With Massive Uncertainty*. RAND, Santa Monica, CA.

Davis, P.K. and J. H. Bigelow. 1998. *Experiments in Multiresolution Modeling*, RAND, Santa Monica, CA.

Davis, P.K. and M. Carrillo. 1997. *Exploratory Analysis of the Halt Problem: A Briefing on Methods and Initial Insights*, RAND, DB-232, Santa Monica, CA.

Davis, P. K., D. Gompert, R. Hillestad, and S. Johnson, *Transforming the Force: Suggestions for DoD Strategy*, RAND Issue Paper, Santa Monica, CA, 1998.

Davis, P. K., D. Gompert,. and R. Kugler, *Adaptiveness in National Defense: the Basis of a New Framework*, RAND Issue Paper, Santa Monica, CA, 1996.

Davis, P.K., J. H. Bigelow, and J.McEver. 2000. *Effects of Terrain, Maneuver, Tactics, and C4ISR on the Effectiveness of Long Range Precision Fire*, RAND, Santa Monica, CA.

Davis,P.K., J. H. Bigelow, and J. McEver. 1999. *Analytic Methods for Studies and Experiments on Transforming the Force*, RAND, Santa Monica, CA.

Lucas, Thomas. 2000. "The Stochastic Versus Deterministic Argument for Combat Simulations: Tales of When the Average Won't Do," *Military Operations Research*, Vol. 5, No. 3.

McEver,J., P. K. Davis, and J. H. Bigelow. 2000a. *EXHALT: an Interdiction Model for the Halt Phase of Armored Invasions RAND*, Santa Monica, CA.

McEver, J., P. K. Davis, and J.H. Bigelow. 2000b. "Implementing Multiresolution Models and Families of Models: From Entity Level Simulation to Personal-Computer Stochastic Models and Simple 'Repro Models'", *Proceedings of the SPIE*, Vol. 4026.

Morgan, G., and M. Henrion. 1992. *Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis*, Cambridge University Press, Cambridge, Mass.

National Research Council. 1997. *Modeling and Simulation, Volume 9 of Technology for the United States Navy and Marine Corps*, 2000–2035, National Academy Press, Washington, D.C., 1997.

R. Lempert, .M. E. Schlesinger, and S. C. Bankes. 1996. "When We Don't Know the Costs or the Benefits: Adaptive Strategies for Abating Climate Change," *Climatic Change*, Vol. 33, No. 2.

Zeigler, B. P.,T. G. Kim, and H. Praehofer. 2000. *Theory of Modeling and Design* , Academic Press, San Diego.

AUTHOR BIOGRAPHY

PAUL K. DAVIS is a senior scientist and Research Leader at RAND, and a Professor of Public Policy in the RAND Graduate School. He holds a B.S. from the U. of Michigan and a Ph.D. in Chemical Physics from the Massachusetts Institute of Technology. Dr. Davis is a recipient of the Wanner Memorial Award of the Military Operations Research Society. He is a member of the Naval Studies Board under the National Research Council and has served on a number of studies for the Council, the Defense Science Board, and the National Institute for Standards and Technology. His e-mail and web addresses are pdavis@rand.org and www.rand.org/personal/pdavis.

DEFINITION AND MEASUREMENT OF MACHINE INTELLIGENCE

GEORGE N. SARIDIS
Professor Emeritus RPI

1. DEFINITIONS OF MACHINE INTELLIGENCE

Recently, there have been a lot of arguments on the subject of Intelligence for Machines that operate autonomously and knowledgeably in unfamiliar or hazardous environments. A position view is presented herein, that represents the engineering point of view as the so called "Intelligent Machines" that implement it are designed and built by engineers (Task Force of the Control System Society of IEEE, chaired by P. Antsaklis1993}

In the last twenty or so years, a lot of discussions have taken place regarding the meaning of **Intelligence**. The psychologists argue about **human intelligence** to be used as the model, while the computer scientists suggest **artificial intelligence** for the job. All these arguments are based on the intelligence that humans demonstrate in dealing with their every day activities, a concept that is still nebulous and very little understood. The engineers stress the concept **of machine intelligence**.

Human Intelligence is to general and poorly understood to be used as model for Intelligent Machines. Artificial Intelligence, on the other hand, was created to deal with the effort to make computers act like human beings, when making decisions and perform other human like activities (Winston 1977). Finally, engineers developed the concept of Machine Intelligence to represent the properties of autonomous machines created to perform unsupervised anthropomorphic tasks (Saridis 1977).

The theory of Intelligent Machines may be thought of as the result of the intersection of the three major disciplines:

- **Artificial Intelligence**,
- **Operations Research**,
- **Control Theory**.

The reason for this claim has been proven necessary is that none of the above disciplines can produce individually a satisfactory theory for the design of such machines. It is also aimed in establishing Intelligent Controls as an engineering discipline, with the purpose of designing Intelligent Autonomous Systems of the future. It combines effectively the results of cognitive systems research, with various mathematical programming control techniques. The control intelligence is hierarchically distributed according to the **Principle of Precision with Decreasing Intelligence (IPDI)**, evident in all hierarchical management systems. The analytic functions of an Intelligent Machine are implemented by Intelligent

Controls, using Entropy as a measure. Such an architecture is analytically implemented using entropy as a measure. However, various cost functions expressed in entropy terms, may be used to evaluate the generated design. Reliability, a property highly desirable for systems functioning autonomously, is a very desirable measure of performance and can also be expressed by entropy, and can be combined in the criterion of performance of the system design.

Intelligence is defined according to the *American Heritage Dictionary (1992) as* :

- **Intelligence (Human) is defined as the capacity to acquire and apply Knowledge.**

Such a statement implies that knowledge is the key variable in an Intelligent system.

The following two definitions, due to P. H. Winston and G. N. Saridis respectively are given to clarify the subject (Winston 1977).

Artificial Intelligence is represented as a mapping of anthropomorphic tasks into the analytic tools of the computer in order to study human behavior, while Machine Intelligence is the inverse mapping of analytic tools imbedded in a machine into anthropomorphic tasks.

- **Artificial Intelligence is the study of ideas which enable computers to do the things that make people seem intelligent. Its central goals are to make computers more useful and to understand the the principle, which makes Intelligence possible.**

The key components of Artificial Intelligence are: interactive systems between man and machine, heuristics and expert system exaustive programming (Saridis 1977).

Some definitions regarding Machine Knowledge and Intelligence are appropriate in order to clearly define the field of Intelligent Machines. In order to establish the definition of Machine Intelligence we revisite the *American Heritage Dictionary (1992)* :

- **Intelligence is defined as the capacity to acquire and apply Knowledge.**

Such a statement implies that knowledge is the key variable in an Intelligent system. Since we shall be dealing with **Machine Intelligence** an appropriate definition is necessary:

- **Machine intelligence is defined as the process of analyzing, organizing and converting data into Machine Knowledge.**

The key components of Machine Intelligence are: computer mathematics, cognitive engineering and Intelligent control.

Now it is well understood that:

- **Knowledge is a form of structured Information.**

This is very convenient because an analytic formulation of Intelligent Machines may be developed, using Shannon's Information Theory. Therfore, **Machine Knowledge** is defined as:

- **Machine knowledge is defined to be structured information acquired and applied to remove ignorance or uncertainty about a specific task pertaining to an Intelligent Machine.**

Similarly,

- **The Rate of Machine Knowledge is the flow of Knowledge in an Intelligent Machine.**

Using the above definitions analytic expressions of Machine Knowledge and its Rate are obtained.

Assuming that Machine Knowledge is Information:

$$K = - \ln[p(K)] \tag{1.1}$$

and the average Rate of Knowledge is also:

$$R = - \alpha - \mu \ln[p(R)] \tag{1.2}$$

where $p(\cdot)$ is the probability density of the event. Solving for $p(R)$ we obtain:

$$p(R) = \exp(- \alpha - \mu R) \tag{1.3}$$

$$\alpha = \ln \int_{\Omega x} \exp(-\mu R)\, dx$$

Complexity is always imbedded in the design and execution of Intelligent Machines. However, their performance is always prescribed by a certain level of detail required by the task expected to be executed, which is defined as **Precision**. Such details are inversely associated with the uncertainty of execution and thus are measurable with entropy. The following definitions help to clarify this concept.

- **Precision is the compliment of the uncertainty of execution of the various tasks of an Intelligent Machine, and Imprecision serves as a measure of the complexity of the process.**

The concept of precision will therefore be associated with the Principle of Increasing Precision and Decreasing Intelligence. Precision is required for the smooth and accurate execution of tasks associated with world processes.

This generalization is found useful in order to accommodate unconventional systems that are served by Intelligent Machines, like biological, environmental etc (Prigogine 1980).

Intelligent Control is the main tool to implement Intelligent Machines. In order to properly implement the Theory of Intelligent Machines the broader definition of Automatic Control Systems is used.

- **Control is making a Process do what we want it to do.**

The Theory of Hierarchically Intelligent Controls has been recently reformulated by Saridis(1996) to incorporate new architectures that are using Neural and Petri nets. The analytic functions of Hierarchically Intelligent Machines are implemented using Entropy as a measure. The resulting hierarchical control structure is based on the Principle of Increasing Precision with Decreasing Intelligence (IPDI) which is discussed in the next Chapter. Each of the three levels of the Intelligent Control is using different architectures, in order to satisfy the requirements of the Principle:

The **Organization level**
modeled after a Boltzmann machine for abstract reasoning, task planning and decision making;

The **Coordination level**
composed of a number of Petri Net Transducers supervised by a dispatcher for command management, serving as an interface with the Organization level;

The **Execution level,**
includes the sensory, navigation and control hardware which interacts one-to-one with the appropriate Coordinators, while a VME bus provides a channel for database exchange among the several devices.

This system was implemented on a robotic tele-transporter, designed for construction of trusses for the Space Station Freedom, at the Center for Intelligent Robotic Systems for Space Exploration laboratories at the Rensselaer Polytechnic Institute.

The basic concepts underlining the theory of Hierarchically Intelligent Machines like Machine Intelligence, Machine Knowledge, Precision and Complexity were defined and contrasted to Artificial and Human Intelligence. The basic difference being the search for an analytic formulation that would lead to an engineering implementation. Further more it has been recently realized that other scientific disciplines have being using the same concepts for an analytic representation of their subjects (Prigogine 1980). Such ideas will be discussed in the next Chapter.

2. THE ENTROPY CONCEPT

Entropy is a form of lower quality energy, first encountered in Thermodynamics. It represents an undesirable form of energy that is accumulated when any type of work is generated. Recently it served as a model of different types of energy based resources, like transmission of information, biological growth, environmental waste, etc. Entropy was currently introduced, as a unifying measure of performance of the different levels of an Intelligent Machine by Saridis (1985). Such a machine is aimed at the creation of **modern intelligent machines which may perform human tasks with minimum interaction with a human operator**. Since the activities of such a machine are energy related, entropy may easily serve as a cost measure of performing various tasks as Intelligent Control, Image Processing, Task Planning and Organization, and System Communication among diversified disciplines with different performance criteria. The model to be used is borrowed from Information Theory, where the uncertainty of design is measured by a probability density function over the appropriate space, generated by **Jaynes' Maximum Entropy Principle.**

Other applications of the Entropy concept are for defining Reliability measures for design purposes and obtaining measures of complexity of the performance of a system, useful in the development of the theory of Intelligent Machines.

Entropy is a convenient global measure of performance because of its wide applicability to a large variety of systems of diverse disciplines including waste processing, environmental, socio-economic, biological and other. Thus, by serving as a common measure, it may expand system integration by incorporating say societal, economic or even environmental systems to engineering processes.

The concept of **Entropy** was introduced **in Thermodynamics** by Clausius in 1867, as the low quality energy resulting from the second law of Thermodynamics. This is the kind of energy which is generated as the result of any thermal activity, at the lower thermal level, and is not utilized by the process.

It was in 1872, though, that Boltzmann used this concept to create his **theory of statistical thermodynamics**, thus expressing the uncertainty of the state of the molecules of a perfect gas. The idea was created by the inability of the dynamic theory to account for all the collisions of the molecules, which generate the thermal energy. Boltzmann (1872) stated that the entropy of a perfect gas, changing states isothermally, at temperature T is given by;

$$S = - k \int_X (\psi-H)/kT \exp\{(\psi-H)/kT\} dx \qquad (2.1)$$

where ψ is the Gibbs energy, $\psi = - kT \ln \exp\{-H/kT\}$, H is the total energy of the system, and k is Boltzmann's universal constant. Due to the size of the problem and the uncertainties involved in describing its dynamic behavior, a probabilistic model was assumed where the Entropy is a measure of the molecular distribution. If p(x) is defined as the probability of a molecule being in state x, thus assuming that,

$$p(x) = \exp\{(\psi - H)/kT\} \qquad (2.2)$$

where p(x) must satisfy the "incompressibility" property over time, of the probabilities, in the state space X, e.g.;

$$dp/dt = 0 \qquad (2.3)$$

The incompressibility property is a differential constraint when the states are defined in a continuum, which in the case of perfect gases yields the Liouville equation. Substituting eq.(2.2) into eq.(2.1) the Entropy of the system takes the form,

$$S = - k \int_X p(x) \ln p(x)\, dx \qquad (2.4)$$

The above equation defines Entropy as a measure of the uncertainty about the state of the system, expressed by the probability density exponential function of the associated energy.

Actually, the problem of describing the entropy of an isothermal process should be derived from the Dynamical Theory of Thermodynamics, considering heat as the result of the kinetic and potential energies of molecular motion. It is the analogy of the two formulations that led into the study of the equivalence of entropy with the performance measure of a control system. If the Dynamical Theory of Thermodynamics is applied on the aggregate of the molecules of a perfect gas, an Average Lagrangian I, should be defined to describe the average performance over time of the state x of the gas,

$$I = \int_{t0}^{tf} L(x,t)\, dt \qquad (2.5)$$

where the Lagrangian L(x,t) = (Kinetic energy) - (Potential energy). The Average Lagrangian when minimized, satisfies the **Second Law of Thermodynamics**. Since the formulations eqs.(2.1) and (2.5) are equivalent, the following relation should be true;

$$S = I/T \qquad (2.6)$$

where T is the constant temperature of the isothermal process of a perfect gas (Lindsay and Margenau, 1957). This relation will be the key in order to express the performance measure of the control problem as Entropy.

In the 1940's Shannon (1963), using Boltzmann's idea, e.g., eq. (3.4), defined **Entropy** (negative) as a measure of the **uncertainty of the transmission of information**, in his celebrated work on **Information Theory**:

$$H = - \int_\Omega p(s) \ln p(s)\, ds \qquad (2.7)$$

where p(s) is a Gaussian density function over the space Ω of the information signals transmitted. The similarity of the two formulations is obvious, where the uncertainty about the state of the system is expressed by an exponential density function of the energy involved.

Shannon's theory was generalized for dynamic systems by Ashby (1965), Boettcher and Levis (1983), and Conant (1976) who also introduced various laws which cover information systems, like the Partition Law of Information rates.

The **ε-entropy** formulation of the metric theory of complexity, originated by Kolmogorov (1956) and applied to system theory by Zames (1979) is another use of entropy.

It implies that an increase in knowledge about a system, decreases the amount of ε-entropy which measures the uncertainty (complexity) involved with the system.

$$\varepsilon - H = \ln(n_e) \qquad (2.8)$$

where n_e is the minimum number of coverings of a set e. Therefore ε-entropy is a measure of complexity of the system involved. It may also be interpreted as a measure of precision if it viewed as the number of points required to describe a line.

Since the latest major improvements in the average quality of life, major increases have occurred in the production of waste, traffic congestion, biological pollution and in general social and environmental decay(Bailey 1990, Brooks Wiley1988, Prigogine1996, Rifkin 1980), which can be interpreted as the increase of the **Global Entropy of our planet** (Saridis 1998), an energy that tends to deteriorate the quality of our modern society. According to the second axiom of thermodynamics this is an irreversible phenomenon, and nothing can be done to eliminate it.

In an attempt to generalize the principle used by Boltzmann and Shannon to describe the uncertainty of the performance of a system under a certain operating condition, Jaynes (1957) formulated his **Maximum Entropy Principle**, to apply it in Theoretical Mechanics. In summary it claims that

- **The uncertainty of an unspecified relation of the function of a system is expressed by an exponential density function of a known energy relation associated with the system.**

As an exampıe of the use of Entropy as a measure of performance, is a modified version of the Principle, as it applies to the Control problem, is derived in the sequel, using Calculus of Variations (Saridis 1987). The proposed derivation represents a new formulation of the control problem, either for deterministic or stochastic systems and for optimal or non-optimal solutions.

The purpose of this work is to establish entropy measures, equivalent to the performance criteria of the optimal control problem, while providing a physical meaning to the latter. This is done by expressing the problem of control system design probabilistically and assigning a distribution function representing the uncertainty of selection of the optimal solution over the space of admissible controls. By selecting the worst case distribution, satisfying **Jaynes' Maximum Entropy Principle**, the performance criterion of the control is associated with the entropy of selecting a certain control (Jaynes 1957, Saridis 1985). Minimization of the differential entropy, which is equivalent to the average performance of the system, yields the optimal control solution. Furthermore, the Generalized Hamilton-Jacobi-Bellman equation is derived from the incompressibility over time condition of the probability distribution. Adaptive control and stochastic optimal control are obtained as special cases of the optimal formulation, with the differential entropy of active transmission of information, claimed by Fel'dbaum (1965), as their difference. Upper bounds of the latter may yield measures of goodness of the various stochastic and adaptive control algorithms. In this section, the entropy measure for optimal control will be established.

The optimal feedback deterministic control problem with accessible states is defined as follows: given the dynamic system;

$$dx/dt = f(x,u,t) \; ; \; x(t_0) = x_0;$$

and the cost function,

$$V(u;x_0,t_0) = \int_{t0}^{T} L(x,u,t)\, dt \qquad (2.9)$$

where $x(t)\varepsilon\Omega_x$ is the n-dimensional state vector $u(x,t)\varepsilon\Omega_u XT \subset \Omega_x XT$, is the m-dimensional feedback control law and $t \; \varepsilon \; \mathfrak{S} = [t_0,T]$.

An optimal control $u^*(x,t)$ is sought to minimize the cost,

$$V(u^*;x_0,t_0) = \underset{u}{\mathrm{Min}} \int_{t0}^{T} L(x,u,t)\, dt \qquad (2.10)$$

Define the differential entropy, for some u(x,t),

$$H(x_0,u(x,t),p(u)) = H(u) = -\int_{\Omega x0}\int_{\Omega x} p(x_0,u) \ln p(x_0,u)\, du dx_0 \qquad (2.11)$$

where $x_0 \in \Omega_{x0}$, $x \in \Omega_x$ the spaces of initial conditions and states respectively, and $p(x_0,u)=p(u)$ the probability density of selecting u. One may select the density function p(u) to maximize the differential entropy according to **Jaynes' Maximum Entropy Principle** (Jaynes 1957), subject to $E\{V(x_0,u,t)\}=K$, for some u(x,t). This represents a problem more general than the optimal where K is a fixed but unknown constant, depending on the selection of u(x,t).

For appropriate constants λ and μ, the worst case density is,

$$p(u) = e^{-\lambda-\mu V(u(x,t),x0,t0)}$$

$$e^{\lambda} = \int_{\Omega x} e^{-\mu V(u(x,t),x0,t0)}\, dx \qquad (2.12)$$

and the total Entropy is equivalent to the average cost function:

$$H(u) = \lambda + \mu E\{V(u(x,t),x_0,t_0)\} \qquad (2.13)$$

and the corresponding minimum value with respect to u(x,t) represents the optimal design.

In most organization systems, the control intelligence is hierarchically distributed from the highest level which represents the most intelligent manager to the lowest level which represents the worker, which is a manifestation of the Principle of **Increasing Precision with Decreasing Intelligence (IPDI)**,. On the other hand, the precision (complexity) or skill of execution is distributed in an inverse manner from the bottom to the top as required for the most efficient performance of such complex systems. This has been analytically formulated as the **Principle of Increasing Precision with Decreasing Intelligence (IPDI)**, by Saridis (1989). The formulation and proof of the principle is based on the concept of Entropy in that report.

According to the IPDI Principle:

- **Machine Intelligence (MI) is the set of actions and/or rules which operates on a Data-base (DB) of events or activities to produce flow of knowledge.**

$$\mathbf{(MI) : (DB) \Rightarrow (R)} \qquad (2.14)$$

This principle suggests that for constant flow of knowledge through the machine less intelligence more data (complexity) are required. Thus it provides an interesting definition of **Machine Intelligence** that has been debated. This has been realized in the three level architecture of Intelligent Machines discussed in the previous section. In the case of fixed database DB, a measure of Machine Intelligence being the Entropy of Knowledge flow, may be concluded:

$$E\{MI\} = H(R) \qquad (2.15)$$

3. CONCLUSIONS

A set of definitions leading to the concept of **Machine Intelligence** have been discussed in this paper, and it is contrasted to **Artificial and Human Intelligence**. Entropy, defined as a Universal Energy, resulting from the production of Work in a system, may successfully serve as a measure of **Machine Intelligence**. The production of Entropy is irreversible without the use of additional work, and may represent thermal energy in Thermodynamics, Information in Communication systems, Performance in Control systems, as well as waste and pollution in Ecological systems, Economic spending in Societal systems, or Biodegradation in Biological systems (Bailey 1990, Boltzmann 1872, Brooks Wiley1988, Prigogine1996, Shannon 1963, Saridis 1998, Rifkin 1980). It represents an unifying measure for globalization of many, up to now, disjoint sciences and may successfully be used as measure for the development of Hierarchically Intelligent Machines with the use of the **Principle of Increasing Precision with Decreasing Intelligence.**

REFERENCES

Antsaklis, P., et al (1993), *Final Report of the Task Force on Intelligent Control* ***IEEE Control Systems Society Magazine*** December.

Antsaklis P. Chair (1994), "Defining Intelligent Control" Report of the Task Force on Intelligent Control, ***IEEE Control Systems Magazine*** Vol. 14, No. 3, p. 4.

Ashby W. R., (1975), ***An Introduction to Cybernetics***,J. Wiley & Sons, Science Edition, New York.

Bailey K. D., (1990), ***Social Entropy Theory,*** State University of New York Press, Albany N.Y.

Boettcher K. L., Levis A. H., (1983),"Modeling the Interacting Decision-Maker with Bounded Rationality", *IEEE Transactions on System Man and Cybernetics*, ***Vol. SMC-12***, 3, pp.

Boltzmann L. (1872), "Further Studies on Thermal Equilibrium Between Gas Molecules", Wien Ber., ***Vol. 66***, p. 275.

Brooks D. R. and Wiley, E. O. (1988) ***Evolution as Entropy*** University of Chicago Press, Chicago Il.

Conant, R. C., (1976), "Laws of Information which Govern Systems", *IEEE Transactions on System Man and Cybernetics*, ***Vol. SMC-6***, No. 4, pp. 240-255.

Faber M., Niemes N., Stephan G., (1995), ***Entropy, Environment and Resourses,*** Springer-Verlag Berlin Germany.

Feld'baum, A.A. (1965), ***Optimal Control Systems***, Academic Press, New York.

Jaynes, E.T. (1957), "Information Theory and Statistical Mechanics", *Physical Review*, ***Vol.4***, pp. 106.

Kolmogorov, A.N. (1956), "On Some Asymptotic Characteristics of Completely Bounded Metric Systems", *Dokl Akad Nauk*, SSSR, ***Vol. 108***, No. 3, pp. 385-389.

Lindsay, R.B., Margenau, (1957), ***Foundations of Physics***, Dover Publications, New York NY.

McInroy J.E., Saridis G.N.,(1991), "Reliability Based Control and Sensing Design for Intelligent Machines", in ***Reliability Analysis*** ed. J.H. Graham, Elsevier North Holland, N.Y.

Prigogine, I., (1980), ***From Being to Becoming***, W. H. Freeman and Co.
San Francisco, CA.

Prigogine, Ilya, (1996), ***La Fin des Certitudes*** Editions Odile Jacob, Paris France.

Rifkin, Jeremy, (1989) ***Entropy into the Greenhouse World*** Bantam Books New York.

Saridis, G.N. (1979), "Toward the Realization of Intelligent Controls", *IEEE Proceedings*, ***Vol. 67***, No. 8.

Saridis, G. N. (1983), "Intelligent Robotic Control", *IEEE Trans. on Automatic Control*, ***Vol. 28***, No. 4, pp. 547-557, April.

Saridis, G.N. (1985), "An Integrated Theory of Intelligent Machines by Expressing the Control Performance as an Entropy", *Control Theory and Advanced Technology*, ***Vol. 1***, No. 2, pp. 125-138, Aug.

Saridis, G.N. (1988), "Entropy Formulation for Optimal and Adaptive Control", *IEEE Transactions on Automatic Control*, ***Vol. 33,*** No. 8, pp. 713-721, Aug.

Saridis, G.N. (1989), "Analytic Formulation of the IPDI for Intelligent Machines", ***AUTOMATICA the IFAC Journal, 25***, No. 3, pp. 461-467.

Saridis, G. N. (1995) ***Stochastic Processes, Estimation, and Control: The Entropy Approach,*** John Wiley and Sons, New York.

Saridis, G.N., (1996)," Architectures for Intelligent Controls" *Chapter 6*, in ***Intelligent Control Systems***, M. M. Gupta, N. K. Singh (eds) IEEE Press New York NY.

Saridis, G. N., (1998), “Optimal Control of Global Entropy for Environmental Systems”***IEEE Robotics and Automation Magazine***, Vol. 5, No. 3, Sept.

Saridis, G.N. and Graham, J.H. (1984), "Linguistic Decision Schemata for Intelligent Robots", ***AUTOMATICA the IFAC Journal, 20***, No. 1, pp. 121-126, Jan.

Shannon, C. and Weaver, W. (1963), *The Mathematical Theory of Communications*, Illini Books.

Valavanis, K.P.,Saridis, G.N., (1992), ***Intelligent Robotic Systems: Theory and Applications,*** Kluwer Academic Publishers, Boston MA.

Wang, F., Saridis, G.N. (1990) "A Coordination Theory for Intelligent Machines" ***AUTOMATICA the IFAC Journal, 35,*** No. 5, pp. 833-844,Sept.

Winston P. H. (1984), ***Artificial Intelligence,*** Addison Wesley, Reading MA.

Zames, G. (1979), "On the Metric Complexity of Casual Linear Systems, ϵ-entropy and ϵ-dimension for Continuous Time", *IEEE Trans. Autom. Control, 24*, 2, pp. 220-230, April.

Domain Independent Measures of Intelligent Control

David Friedlander
Shashi Phoha
Applied Research Laboratory
The Pennsylvania State University
University Park, PA 16802

Asok Ray
Mechanical Engineering Department
The Pennsylvania State University
University Park, PA 16802

ABSTRACT

There is no standard method for measuring intelligence in artificial systems. One reason for this is that no single definition of intelligence exists. Another is that many of the definitions of intelligence are not appropriate for artificial systems based on the current level of scientific understanding. This includes *introspective*, as opposed to *behavioral*, measurements. This paper explores quantitative, domain independent measures of intelligence for discrete event control systems. It is motivated by traditional measures of effective control such as *controllability*, and *robustness*, and includes original work on *robustness* and *permissiveness*.

KEYWORDS: *intelligence, artificial systems, control systems, hierarchical control, intelligent control*

1. INTRODUCTION

1.1. INTELLIGENCE OF ARTIFICIAL SYSTEMS

A single Intelligence Quotient, as generated by a standardized IQ test, has been used as a measure of general human intelligence. More recent work suggests that there are multiple types of intelligence. This concept is essential to the development of intelligence measurements for artificial systems for two reasons. First, current systems have not reached the level where they can display behavior indicative of a nontrivial understanding of the general representations of human knowledge such as language or mathematics. Second, current techniques in constructing artificial systems result in a sharp trade-off between the quality of performance and the breadth of the domain. In order to perform at a reasonable level, most artificial systems are restricted to a relatively narrow domain.

The current criteria for intelligent systems tend to look at either how well the system performs its assigned tasks, or to what extent can the system behave in ways that are characteristic of human intelligence. The former criteria tend to be domain specific, implementation independent, and relatively easy to quantify and measure. The latter criteria tend to be domain independent, implementation dependent, and are difficult to quantify and measure.

Domain independent intelligent behaviors that can be approximated in current artificial systems include:

- reacting effectively to novel stimuli and situations,
- perceiving essential properties from a large, complex world of sensory information,
- identifying and taking action on the essential problem of a given situation,
- making appropriate decisions in a variety of situations, in complex environments,
- recognizing and exploiting opportunities within one's environment,
- recognizing patterns within the environment,
- manipulating symbols,
- overcoming obstacles,
- correcting for errors,
- handling uncertainty,
- and learning from experience.

These behaviors are difficult to measure in the general case. This paper presents quantitative measurements of intelligent behaviors for systems based on hierarchical networks of discrete event controllers. Some of the metrics used here are based on extending methods from continuous control to discrete event control systems.

The methods are illustrated with two types of examples. The first is a highly simplified control system for command and control (C^2) of aircraft operations in battle management as illustrated in Figure 4. The second is an abstract controller of slightly greater complexity. The lowest level is shown in Figure 1; hierarchical versions are shown in Figures 2 and 3.

This paper examines the following characteristics of hierarchical control systems:

- *Controllability*, the ability of the system to accomplish its goals without reaching an error state from which it cannot recover.
- *Hierarchical Consistency*, the ability of a higher-level controller to achieve its goals indirectly, by controlling one or more lower-level controllers.
- *Robustness*, the ability of the system to operate under uncertainty and novel situations, and to recover from errors,

- *Permissiveness*, the ability to achieve goals through more than one path.
- *Aggregation*, the ability to abstract essential properties from lower level data,
- *Disaggregation*, the ability to take effective specific actions based on higher level abstractions,
- *Scalability*, the ability to handle large, complex environments, and

1.2 Hierarchical Finite State Automaton (FSA) control systems

Although, computational theory shows that finite state automata (FSA) are more restricted than more general representations such as general finite state machines, it has been shown that networks of FSAs do have the same computational power as general finite state machines [5] such as digital computers.

Control systems have a history of successfully operating large complex systems of interacting components in real-time. For most of this history, control systems were *continuous*. They used sensors to collect data from various points in the system (often referred to as the *plant*), process the data, and perform actions on the plant through the use of mechanical devices. A number of metrics such as controllability and robustness have been developed to measure the performance of the control system in controlling the plant.

The concept of a continuous control system was extended to discrete event systems (DES) by Ramadge and Wonham [7]. Instead of processing continuous numerical data, DES controllers process strings of symbols that form a formal language. This has resulted in a synergy with work on processing formal languages from the discipline of Computer Science, and resulted in mathematical theorems that provide ample insight to the strengths and limitations of the approach. Artificial Intelligence approaches do not have the same extent of mathematical grounding.

In DES control, the plant can be considered as a machine that processes symbols from a (finite) alphabet in a way that forms a formal language, L, over an alphabet of symbols, Σ. The alphabet, Σ, can be partitioned into symbols corresponding to uncontrollable events, Σ_u, and symbols corresponding to controllable events, Σ_c. These symbols represent events in the system. An example of a controllable event is a friendly (controlled) aircraft firing at an enemy target, while an example of an uncontrollable event is the enemy target firing on the friendly aircraft. The controller can be thought of as a recognizer of uncontrollable events and a generator of controllable ones. In analogy to continuous control systems, the controllable events are the plant input (feedback), and the uncontrollable events are the plant output.

The association of DES controller actions with decisions expands the notion of intelligent control towards general intelligence. Performance measurements from continuous control theory have been extended to include DES control. New measurements, which are mathematically grounded, have also been developed for DES controller performance. DES control systems can exhibit intelligent behavior in the same way as AI software or robots. The performance measures for intelligent DES control are an indication of intelligence in the more generic sense.

More recent work has extended the notion of single DES controllers to interacting hierarchies of DES controllers. This includes mathematical work to extend the notion of controllability to hierarchies, where it is called *hierarchical consistency* [13]. The use of a hierarchy allows the controller to make complex decisions while taming the explosion of the number of states that would take place in a single controller performing the same function. Lower level controllers transmit an *aggregated* version of their formal languages to higher-level controller nodes. The higher-level nodes exert control on lower levels by enabling and disabling controllable events at the lower levels. Such systems exhibit behavior analogous to forming conclusions and formulating and executing plans.

2. Controllability and Hierarchical Consistency

The controllable events of the system can be disabled, i.e., prevented from occurring, by the controller. It uses this ability to influence the evolution of the system by preventing certain events from occurring at certain times.

The goals of the control system are given as a set of specifications such as, "If the aircraft runs out of weapons, it returns to base." Given the uncontrolled language of the plant, L, the specifications can be formalized as a sublanguage of L, $K \subseteq L$. The controller attempts to enforce the specifications by restricting the plant to operate within K, rather than L. The plant is said to be *controllable* by the controller if it can operate the plant in a way that satisfies the specifications.

A controllable system is able to follow its specifications from any of the system states. Even when an uncontrollable event moves the system in an unanticipated way, there is a path that satisfies the specifications. This shows some characteristics of purposeful behavior.

FSA controllers can be organized in a hierarchy where high-level controllers achieve high-level goals by observing and controlling low-level controllers. *Hierarchical consistency*, the hierarchical equivalent of controllability, is the ability of the high-level controller to achieve its goals in this way, starting from any legal combination of states in the high and low level controllers. This behavior, when it can be achieved, shows the ability of the system to achieve high-level goals through low-level actions, suggesting an ability to formulate and carry out plans.

2.1. *Controllability*

The controllability definition can be formalized as: K is controllable if $\overline{K}\Sigma_u \cap L \subseteq \overline{K}$, where $\overline{K}$ is the prefix closure language of K. That is, the occurrence of any uncontrollable event at any time permitted by the dynamics of the plant, will not violate the control specifications.

Figure 1 shows how a controller would be implemented in practice. The controller has been implemented as a finite state machine with the specification that the plant should always return to state S_0. The first event, a, moves the controller from state S_0 to state S_1 and the second event, b, moves it from state S_1 to state S_2. At state S_2, the controller takes action, e, and moves to state S_4, where it takes action h, and returns to state S_0. The system's performance is characterized by the string a,c,e,h, which is in $\overline{K}$ if the plant is controllable by the controller.

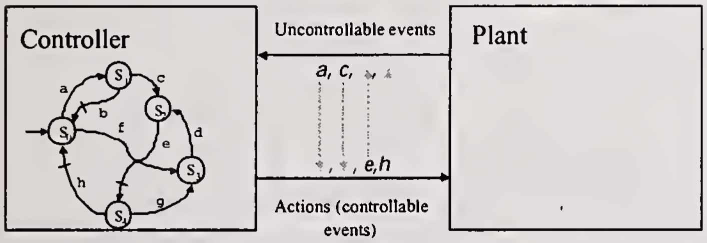


Figure 1. Discrete Event System Control

The example exhibits behavior that looks purposeful. It seeks to bring the plant back to a goal state S_0. It also gives the appearance of overcoming obstacles, uncontrollable events, in an uncertain environment. For example it could not perform action e when it was in state S_1, because event c occurred. It then achieved its goal in another way, by taking actions e and h. If the plant is controllable, there will always be a path to the goal state.

In a large and complex enough system, these behaviors would appear intelligent. These types of systems have been implemented to solve complex applications such as controlling an automated factory.

2.2. *Hierarchical Consistency*

A hierarchical structure is used in control of dynamic systems for a variety of tasks. Control is divided between higher levels, which process events of greater generality and larger scope; and lower levels, which process more specific events of lesser scope.

This notion has been formalized in a way that is illustrated in Figure 2. The higher level controller is designed to control a virtual high level plant with the following components: the low-level, i.e. *actual*, plant; the low-level controller, M, a mapping from strings of the low-level plant language to high-level events; and U, a mapping from high-level, controllable events to low-level *control patterns*. A control pattern is a set of low-level controllable events that are to be disabled in the low-level controller.

We start with a low-level plant with language L_p, and a low-level controller with language $K_{lo} \subseteq L_p$. We wish to implement further restrictions on the performance of the low-level plant to a language $\widetilde{K}_{lo} \subseteq K_{lo} \subseteq L_p$. This is to be done by translating strings from L_p to a high level language, $L_{hi} = M(L_p)$, which has an alphabet that containing controllable and uncontrollable symbols, $\Sigma^{hi} = \Sigma_u^{hi} \cup \Sigma_c^{hi}$. In the example, each state in the low-level controller is labeled with either a high-level symbol or t_0. This implieitly determines M. Whenever the low-level controller enters a state marked with a high-level symbol, the symbol is added to the high-level translation of the low-level string.

In order for this scheme to work, the high-level controller needs to be able to control the virtual high-level plant via the mechanism shown in Figure 2. This is called *hierarchical consistency*. It is true when $M(\widetilde{K}_{lo}) = K_{hi}$.

Note that the example is hierarchically consistent because

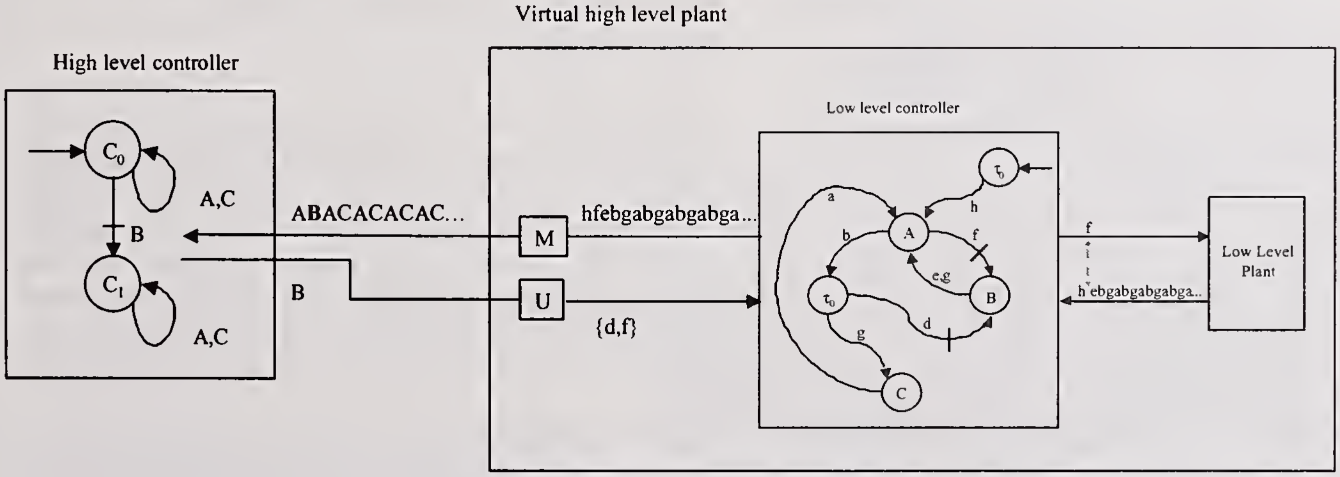


Figure 2. Hierarchical Control

the only controllable high-level event, *B*, can be disabled with the its associated control pattern {d,f}, which blocks the low-level controller from entering the state which transmits *B* to the high-level controller. This is done by disabling all of the low-level (controllable) events leading to thc low-level state marked *B*.

The high-level controller adds the following requirement to the behavior of the low-level controller: *an event from the set {d,f} can only occur once in any low-level string.* This translates to the high-level requirement that: *the event* B *can only occur once in any high-level string.*

3. Robustness and Permissiveness

Two additional concepts in DES control theory that relate to intelligent behaviors are *robustness* and *permissiveness*. Robustness is the ability of the controller to handle uncertainty. In this section we will look at robustness with respect to uncertainty in the characteristics of the actual plant. Permissiveness is the ability of the controller to allow a wide range of behaviors while operating the plant within the control specifications. This behavior gives the appearance of resourcefulness. It may allow the system to satisfy its requirements in multiple ways and might improve performancc in the real world where a single path to a goal may be blocked by an unanticipated event.

In this section, we first propose a measure for formal languages and then use it to derive quantitative measures of robustncss and permissiveness.

3.1. Robustness

In reality, the actual plant is not completely known. It is represented by an FSA, called the nominal plant model, which contains all of the available knowledge about the actual plant. The language of the plant is approximated by the language of the nominal plant model, and the controller is designed on this basis.

It is therefore important to determine whether the controller can control languagcs (i.e., plants) other that the one for which it was designed. This quantity is known as *robustness*. The more robust the controller, the more likely it is to be able to control the actual plant.

We have determined a method to estimate robustness through the development of two concepts, a population of plant languages near the language of the nominal plant model, and a measure for formal languages of a given alphabet, Σ.

The population of plant languages can be defined as the languages of plants derived from the nominal plant model through the application of a small number of primitive operations such as the addition or deletion of a single state or transition, under the restriction that the results define a deterministic FSA.

The measure of formal languages can be defined in terms of a weighted partition of the set Σ^* of all finite strings in the alphabet Σ. The countable set Σ^* is partitioned into disjoint subsets, S_i, such that $\Sigma^* = \bigcup_{i=1}^{n} S_i$, where *n* is at most countable infinity (i.e., either *n* is a finite positive integer or infinity). Each subset is assigned a weight, w_i such that $\sum_1^n w_i = 1$. The measure of a given language *L*, is defined as:

$$\mu(L) = \sum_1^n w_i \Delta_i(L), \tag{1}$$

$$\text{where } \Delta_i(L) = \begin{cases} 1 \; if\; \exists s \in L \; such\, that\; s \in S_i \\ \quad 0 \; otherwise \end{cases}$$

The robustness of a controller, *C*, can then be defined as: $R(C) = \sum_{P_i \in R} M(L(P_i) - L(P)) \cdot D(L(P_i))$, where *R* is the population of plant models related to *P*, the nominal plant model; and *D*(*L*) = 1 if *C* can control *L*, 0 otherwise.

3.2. Permissiveness

Given a set of specifications in the form of a language, *E*, a controller, K_1, controls a plant, *G*, if $L(G \mid K_1) \subseteq E$. There could be, however, another controller, K_2, such that $L(G \mid K_1) \subseteq L(G \mid K_2) \subseteq E$. In this case, K_2 is considered more *permissive* than K_1. Although both controllers control the plant, K_2 allows a greater range of behaviors in the closed loop system.

Since $L(G \mid K) \subseteq L(G)$, for any controller defined on *G*, it follows that, for any controller *K* which satisfies *E*, $L(G \mid K) \subseteq L(G) \bigcap E$. Using the proposed language measure, we can define the permissiveness of a controller, *K*, defined on a plant, *G*, operating within the specification language, *E*, as

$$P(C) = \left(\frac{\mu(L(G \mid K))}{\mu(L(G) \bigcap E)} \right) \in [0,1]. \tag{2}$$

4. Aggregation, Disaggregation, and Scalability

The technique for hierarchical control, described in Section 2, can be extended to allow a high-level controller to control more than one low-level plant. It can also be extended to form multilevel hierarchies of arbitrary size. The number of nodes in the entire hierarchy grows linearly with the number of leaf nodes. Since the leaf nodes recognize events and take actions on the plant (i.e., the outside world), the coordination provided by hierarchical control networks is *scalable*.

There is a mapping from events in the low-level, child controllers to events in the higher-level, parent controller, and a corresponding mapping from controllable events in the parent controller to control patterns in the child controllers. If the mapping from child to parent compresses the data, the

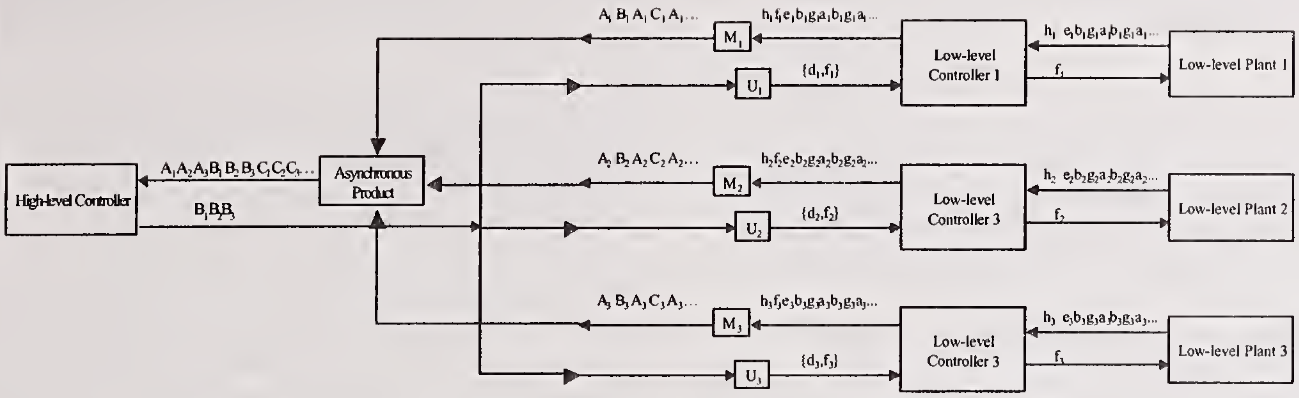


Figure 3. Hierarehieal Control of Multiple Low-level Controllers

control strueture will exhibit some additional intelligent behaviors. It will appear to *draw conclusions* from lower level events, make deeisions based on abstraetions, can carry out the decisions at the lower, possible physieal, level. There is currently no technique to synthesize or measure "good" aggregations from lower to higher level strings in formal languages. We will therefore use the data compression ratio as an indication of this ability.

Figure 3 illustrates how a high-level controller can control more than one low-level plant. The events being reeognized by the high-level controller is the asynchronous produet of the high-level symbols being produced in each low-level controller, i.e., they are sent to the high-level eontroller in the order of their occurrence. Multilevel hierarchies are formed whcn every non-root node sends higher-level symbols to the level above it, and every non-leaf node sends control patterns to the level below it.

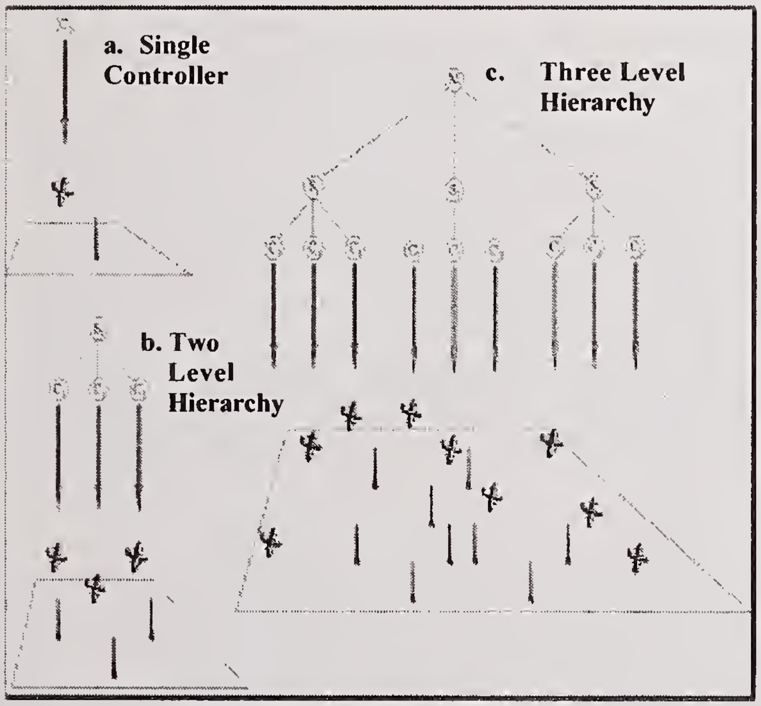


Figure 4. Sealability of DES Controller Hierarchies

The scalability of control hierarchies is shown in Figure 4. The size of the hierarehy increases in proportion to the size of the battlespaee. In 4a, a single eontroller/plane is attacking a single enemy target located in a given area, *A*. In 4b, a two level controller hierarchy with threc planes is attacking three enemy targets in an area of *3A*, and in 4c, a three level controller hierarchy with nine planes is attacking nine enemy targets in an area of size *9A*.

5. Conclusions

Intelligent control, using a hierarchy of discrete event controllers, is a good applieation for deriving quantitative measures of intelligence because these systcms are eomplex enough to cxhibit intelligent behavior, but simple enough to allow for a relatively thorough mathematical analysis. Domain independent, quantitative measures of controller performance derived from the analysis are correlated with intelligent behavior by the system. Controllability and hierarchical consistency are correlated with goal-seeking, purposeful behavior; robustness is correlated with the ability to handle uncertainty, and permissiveness is correlated with resourcefulness. The process of aggregation and disaggregation in hierarchical control suggests the ability to make abstractions, plan, and carry out plans.

6. Acknowledgment

Some of the eoneepts of intelligenee metrics presented in this paper were initially developed on the JFACC projeet, sponsored by the Defense Advanced Research Projeets Agency (DARPA) and Air Foree Research Laboratory, Air Foree Materiel Command, USAF, under agreement number F30602-99-1-0547 (JFACC). The U.S. Government is authorized to reproduee and distribute reprints for Government purposes notwithstanding any copyright annotation thereon. The views and eonclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced

Research Projects Agency (DARPA), the Air Force Research Laboratory, or the U.S. Government.

7. References

[1] Barlow, H. B., *Oxford Companion to the Mind* (1987).

[2] Baum, A., Newman, S., Weinman, J., West, R. and McManus, *Cambridge Handbook of Psychology Health and Medicine*. Cambridge, Cambridge University Press 1997.

[3] Miele, F., "Skeptic Magazine Interview With Robert Sternberg on The Bell Curve," Skeptic vol. 3, no. 3, 1995, pp. 72-80.

[4] Harnad, S., "Minds, Machines and Searle," *Journal of Theoretical and Experimental Artificial Intelligence* 1: 5-25, 1989.

[5] Delorme M. and Mazoyer, J., *Cellular Automata, A Parallel Model,* Kluwer Academic Publishers, The Netherlands, 1999, pp. 32-35.

[6] Ramadge, P.J. and Wondham, W.M., "Supervisory Control of a Class of Discrete-Event Processes," *SIAM J. Contr. Optim.*, Vol. 25, No. 1, pp. 206-230.

[7] Ramadge P.J. and Wondham, W.M., "The Control of Discrete Event Systems," *Proc. IEEE,* Vol. 77, No. 1, January, 1989.

[8] Ray, A., Xi, W., Zang, H.and Phoha, S., "Hierarchical Consistency of Supervisory Command and Control of Aircraft Operations," *Proc. 2nd Symp. on Advances in Enterprise Control,* Minneapolis, MN, July 10-11, 2000, published by IEEE.

[9] Searle, J. R., "Minds, Brains and Programs," *Behavioral and Brain Sciences 3:* 417-424, 1980.

[11] Takai, S. "*Synthesis of Robust Supervisors for Prefix-Closed Language Specifications*," to be published.

[12] Wondham, W.M. *Notes on Control of Discrete-Event Systems. Systems Control Group, Dept. of Electrical and Computer Engineering,* U. of Toronto, April 1999.

[13] Zhong H. and Wondham, W.M. "On the Consistency of Hierarchical Supervision in Discrete-Event Systems," *IEEE Transactions on Automatic Control,* Vol. 35, No. 10.

Choosing Knowledge Granularity for Efficient Functioning of An Intelligent Agent

Yiming Ye[a] and John K. Tsotsos[b]
IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA[a]
Department of Computer Science, York University, Toronto, Ontario, Canada M3J 1P3[b]
(1) 914 784-7460[a]   (1) 416 736-2100[b]
yiming@watson.ibm.com[a]   tsotsos@vis.toronto.edu[b]

Abstract

In this paper we introduce the concept of knowledge granularity and study its influence on an agent's action selection process. The goal is to provide a guideline for an agent to select a reasonable knowledge granularity for a given task. Finally we present an idea of using an adaptive mesh method for uneven granularity representation.

1 Introduction

An agent is a computational system that inhabits dynamic, unpredictable environments. It has knowledge about itself and the world. This knowledge can be used to guide its action selection process when exhibiting goal-directed behaviors. Here we address the following question: "How much detail should the agent include in its knowledge representation so that it can efficiently achieve its goal?" There are two extremes regarding granularity of knowledge representation. At one end of the spectrum is the purely reactive scheme which requires little or even no knowledge representation. At the other end of the spectrum is the purely planning scheme which requires the agent to maintain as much detailed knowledge as possible. Experience suggests that neither purely reactive nor purely planning systems are capable of producing the range of behaviors required by intelligent agents in a dynamic, unpredictable environment. This paper offers an alternative point of view of the spectrum of knowledge abstraction based on the granularity of knowledge representation. The goal is to find the proper balance in representing an agent's knowledge such that the representation is detailed enough for the agent to select reasonable actions, and at the same time it is coarse enough that it does not exhaust the agent's resources when selecting those reasonable actions. At the end of the paper, we propose an idea of using adaptive mesh to represent knowledge within a domain.

2 A Case Study

Here we use object search as an example to study the influence of knowledge granularity on the performance of an agent. Object search is the task of searching for a given object in a given environment by a robotic agent equipped with a pan, tilt, and zoom camera [13]. The goal of the agent is to intelligently control the sensing parameters so as to bring the target into the field of view of the sensor and to make the target in the image easily detectable by the given recognition algorithm. To efficiently detect the target, the agent uses its knowledge about the target position to guide its action selection process. This knowledge is encoded as a discrete probability density that is updated whenever a sensing action occurs. To perfectly encode the agent's knowledge, the size of the cube should be infinitely small - resulting in a continuous encoding of the knowledge. But this will not work in general because an infinite amount of memory is needed. In order to make the system work, the agent is forced to represent the knowledge discretely - to use cubes with finite size. This gives rise to an interesting question: how we should determine the granularity of the representation (the size of the cube) such that the best effects or reasonable effects can be generated. The granularity function **G** can be defined as the total memory used by the agent to represent a certain kind of knowledge divided by the memory used by the agent to represent a basic element of the corresponding knowledge. In this case, **G** equals to the total number of cubes in the environment.

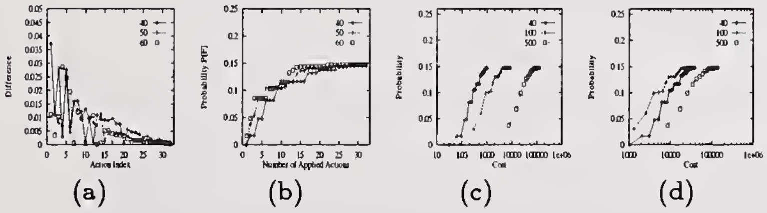


Figure 1: *Experimental results for object search agent.*

We have performed experiments to study the influence of knowledge granularity on the performance of the agent. Usually the higher the value of the knowledge granularity, the longer the time needed to select an action. This is simply because the planning system has more data to be processed. The approximations involved in discretization will cause errors in calculating various values. In general, the higher the value of the knowledge granularity, the less the error caused by discretization. Figure 1(a) shows the errors caused by granularities 40 × 40, 50 × 50, and 60 × 60. The error associated with knowledge granularity may influence the quality of the selected actions, and thus influence the performance of the agent. As shown in Figure 1(b), the higher the granularity, the less the number of actions are needed for the system to reach its detecting limits. Figures 1(c)(d) show the performance of the agent for action execution time 1 second (c) and 1000 second (d), respectively. We can see that a higher granularity may not always beneficial, especially when the action execution time is long.

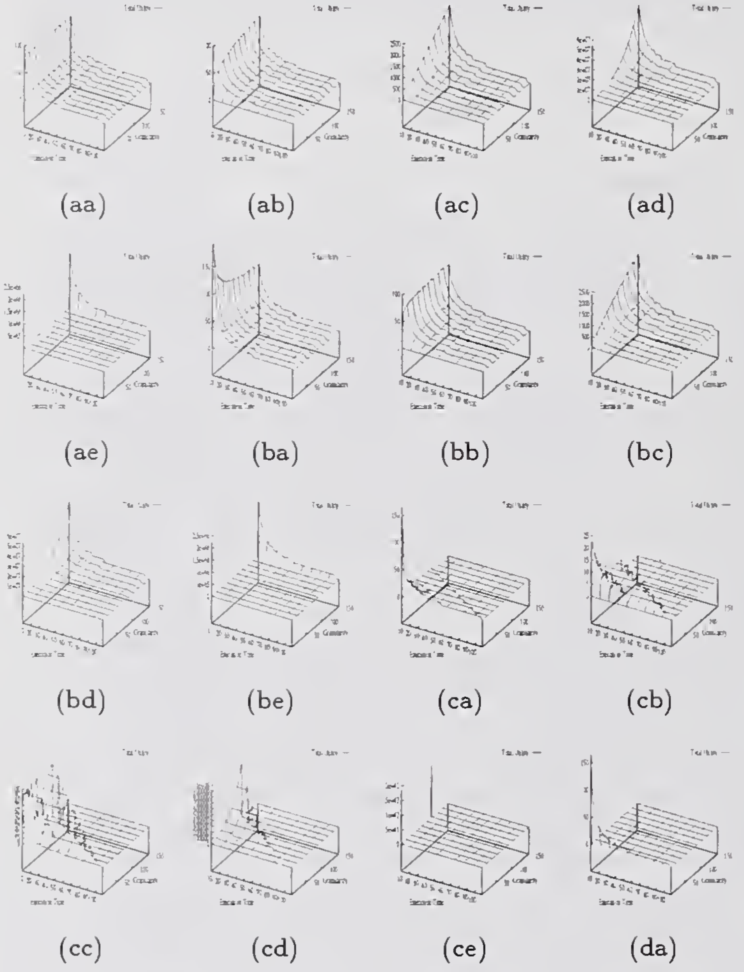


Figure 2: The influence of knowledge granularity on the performance of an agent.

3 Knowledge Granularity in General

In this section, we study in general the influences of knowledge granularity on an agent's action selection performance. For a task oriented agent, a finer granularity usually results in a better selected action. However, the action selection time for a finer granularity is usually longer. Thus, a finer granularity requires more time for selecting actions, and has less time in executing actions. On the other hand, a coarser granularity requires less time for action selection, thus has more time for action execution. In other words, with respect to a fixed time constraint, an agent can usually execute more low quality actions for a coarser granularity, and less high quality actions for a finer granularity. It is thus very interesting to study how the performance of a task oriented agent is influenced by the degree of knowledge granularity, and how the agent should choose a reasonable granularity from the spectrum of knowledge abstraction.

Different agents use different kinds of knowledge and different kinds of action selection procedures. Because of the complexity and diversity of the world of agents, it is impossible to provide a general conclusion or solution with regard to knowledge granularity. What we can do is to group agents into different categories and study the behavior with respect to each category. It is obvious that the performance of an agent is influenced by the action execution time, t_e, the action selection time, t_s, the total time constraint for the given task, T, and the quality Q of the selected and executed actions. t_s and Q is influenced by the knowledge granularity adopted by the agent. Suppose for a granularity g, the *average* time needed in selecting an action is $t_s(g)$, the *average* contributions of a selected action to the task is $Q(g)$. *Assuming* that the total contributions U made by an agent within the time constraint T can be represented by the sum of the average contributions of all the actions that is executed within T. Then, U can be represented as follows.

$$U(g) = \lfloor \frac{T}{t_e + t_s(g)} \rfloor Q(g)$$

In the following, we study how $U(g)$ is influenced by different $t_s(g)$ and $Q(g)$. We assume $T = 100$ and $g \in [1, 150]$.; The following functions are used in our empirical study: $f_a(g) = 5$; $f_b(g) = ln(g)$; $f_c(g) = g + 1$; $f_d(g) = g * g * g + 1$; $f_e(g) = exp(g) + 1$. These functions represent different relations between the knowledge granularity g and the entities to be discussed. Function $f_a(g)$ means that the entity is a constant, and thus is not influenced by granularity. $f_b(g)$, $f_c(g)$, $f_d(g)$, $f_e(g)$ refer to different degrees of the influence of granularity on the entity. Figure 2 shows how the granularity influences the performance of the agent under different situations. The graphs are indexed by the above functions. For example, Figure 2(ab) corresponds to the situations that $t_s(g) = f_a(g)$ and $Q(g) = f_b(g)$.

From Figure 2, we can notice that t_e is a very important factor that influences the selection of the knowledge granularity. Figures 2(aa)(ab)(ac)(ad)(ae) show the situation when t_s is not influenced by the granularity. In this special case, the finer the granularity, the better the performance, except for the first one (aa). From Figures 2 (ca)(cb)(cc)(cd)(ce) we can notice that for a large execution time (≥ 50), g should be low in order to guarantee that at least one action can be executed. Figure 2(ca) shows the situation that the benefit of each action is not influenced by the granularity, thus a smaller granularity is preferred no matter what the action execution time is. From Figure 2(cc) we can notice that the situation becomes complex. For example, for a small t_e, there are several granularities that can generate satisfactory results. These reasonable granularities are different for different t_e.

4 Selecting Knowledge Granularity

The experiments in the above section show that the level of knowledge granularity has a big impact on the quality and speed of the agent's behavior. It is thus important for an agent to adapt its knowledge granularity based on environmental and task-specific demands. In this section, we address the following interesting question: how can we select the knowledge granularity G(k) for a given representation scheme k such that the best agent performance or a relatively good agent performance can be achieved?

In some situations, we are able to select the best knowledge granularity in the sense that it maximizes the performance of the agent. Here is an example. Suppose we have an agent whose task is to collect food from a region of length L within a time limit T. The agent can use different representation lengths $\Lambda = \{l_1, \ldots, l_q\}$ to represent the region. (suppose $\frac{L}{l_i}$ is integer, where $1 \leq i \leq q$). If the agent selects $l \in \{l_1, \ldots, l_q\}$ as its representation scheme k for the corresponding knowledge, then the corresponding knowledge granularity for this scheme will be $\mathbf{G}(\mathrm{k}) = \frac{L}{l}$. The total region is thus divided into $\frac{L}{l}$ units. The process of food collection is as follows. Before the collecting process, all the units of the region will be in the status of "not ready". When the collecting process begin, one of the units becomes "ready". The agent will then search for this unit. The time, $t_s(l)$, used by the agent to locate the unit is the time for the agent to select an action under the current representation scheme. Suppose $t_s(l) = \frac{1}{l}$. After the unit is located, the agent will collect food from this unit. The total time needed for the agent to collect food is the time needed for the agent to execute the selected action. Suppose it is $t_e(l) = Cl$ (where C is a constant). The total amount of food that is collected is $B(l) = \frac{1}{l}$. When the agent finishes its food collection process at the selected unit, the status of another unit will become "ready". The agent will search for this new unit and collect food again from this new unit. This process will continue until the total time T is used up. If the total time T is exhausted when the agent is locating a unit or when the agent is collecting food within a unit, then the amount of collected food from the corresponding unit will be zero. It is obvious that the number of units that can be processed by the agent within T is $\frac{T}{t_s(l)+t_e(l)}$, and the number of units available is $\frac{L}{l}$.

The performance $\mathbf{P}$ of the agent is measured by the total amount of food collected by the agent and is given by the following formula:

$$\mathbf{P} = \begin{cases} \frac{L}{l}B(l) & \text{if } \frac{T}{t_s(l)+t_e(l)} \geq \frac{L}{l} \\ \lfloor\frac{T}{t_s(l)+t_e(l)}\rfloor B(l) & \text{if } \frac{T}{t_s(l)+t_e(l)} < \frac{L}{l} \end{cases} \quad . \tag{1}$$

This is actually

$$\mathbf{P} = \begin{cases} \frac{L}{l^2} & \text{if } l \leq \sqrt{\frac{L}{T-CL}} \\ \lfloor\frac{Tl}{Cl^2+1}\rfloor\frac{1}{l} & \text{if } l > \sqrt{\frac{L}{T-CL}} \end{cases} \quad . \tag{2}$$

The problem is to find a l in $\Lambda = \{l_1, \ldots, l_q\}$ such that $\mathbf{P}$ is maximized. The set Λ can be divided into two parts $\Lambda_A = \{l_1, \ldots, l_j\}$ and $\Lambda_B = \{l_{j+1}, \ldots, l_q\}$, such that all the elements in Λ_A are less than $\sqrt{\frac{L}{T-CL}}$, and all the elements in Λ_B are greater than or equal to $\sqrt{\frac{L}{T-CL}}$. It is obvious that for elements $l \in \Lambda_A$, the smallest one has the best performance because $\frac{L}{l^2}$ is a decreasing function. For elements $l \in \Lambda_B$, we can calculate the value of $\lfloor\frac{Tl}{Cl^2+1}\rfloor\frac{1}{l}$ to identify the best element. Then we compare the smallest element in Λ_A and the best element in Λ_B to identify the one that maximizes the performance of the system.

The above example shows that in some situations, an agent is able to identify an optimum knowledge granularity based on the task requirement (here T) and the environmental characteristics (here L). The basic method is to try to represent the performance of the agent as a function of the agent's knowledge granularity, and then to find the granularity that maximizes the performance.

In general, it is very difficult or even impossible to find a best knowledge granularity for an agent, because the performance of the agent might be influenced by many other factors in addition to the knowledge granularity. For example, there does not exist a best knowledge granularity for the object search agent, because its performance is also influenced by the initial target distribution. A granularity that is best for one distribution might not be the best for another distribution. Thus, in general, we need to relax our requirements. Instead of finding the best granularity, we search for a reasonable one such that a relatively good performance can be achieved. Because of the variations of different agent systems, it is impossible to provide a detailed procedure to select the acceptable granularity that can be applied to all the agent systems. However, we can provide a general guideline for the selection of the knowledge granularity.

5 Selecting Reasonable Granularity in Complex Agent Environment

In an agent environment where the relationships among the task constraints, the environments, and the knowledge granularity are very complex, the "demand-environment-granularity" (DEG) Hash Table can be used to select a reasonable granularity. The DEG Hash Table is a Hash Table such that the "key" is the combination of different factors and the "value" is the granularity that is appropriate for the corresponding factors. When an agent is informed of task requirements, it first transforms the task requirements and the environmental factors into a key. Then it retrieves the granularity from the DEG Hash Table based on the key. This granularity will be used by the agent to represent the corresponding knowledge.

For a complex agent environment, it might have more than one task constraints T_1, ..., T_{n_T}. Each $\mathbf{T}_i$ forms one component in the "key" of the DEG Hash Table. It can be divided into several groups $T_{i,1}$, ..., $T_{i,k}$ based on certain criteria. For example, the task constraint for an object search agent is the total time available for the search. This time constraint can be divided into groups like "from 1 second to 30 seconds", "from 30 seconds to 100 seconds", etc..

In addition to the task constraints, we should also consider the influences of the environmental factors when selecting the granularity. Suppose E_1, ..., E_{n_E} are the environment factors that need to be considered. Like above, each $\mathbf{E}_i$ can be divided into several groups $T_{i,1}$, ..., $T_{i,k}$ based on a certain criteria.

The DEG Hash Table is then looks like following:

$\mathbf{T}_1$	...	$\mathbf{T}_{\mathrm{n_T}}$	$\mathbf{E}_1$	...	$\mathbf{E}_{\mathrm{n_E}}$	$\mathbf{G}$
t_1	...	t_{n_T}	e_1	...	e_{n_E}	g
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 1

Where each row in the table, except the first one, gives a "key" $(t_1, \ldots, t_{n_T}, \mathrm{e}_1, \ldots, \mathrm{e}_{n_E})$ and the corresponding granularity value g. Here, t_i is a category (group) for the task constraint factor $\mathbf{T}_i$ and e_i is a category (group) for the environmental factor $\mathbf{E}_i$. Term g is the knowledge granularity value corresponding to the "key" and should be obtained by conducting various simulation experiments or theoretical analysis before the agent performs any task. When an agent is informed of a task, it first determines the key based on the current situations, and then uses this "key" to locate the knowledge granularity.

6 The Adaptive Mesh Approach

The above analysis and methods assumes that an agent maintains exactly the same granularity for the whole region. This may not be necessary. For example, for the case of object search, the search agent may only need to have a detailed representation of the surrounding area or areas with high density. Because these areas are the most important areas that need to be considered during the sensor planning process. Thus, the granularity should be different for different areas. Actually people seldom maintain the same granularity when they perform tasks. They dynamically adjust the granularity at different time and under different situations. For example, when a person travels from his university to another city to attend a conference, the representation of the geographical situations or maps will be different at different times and context. Before he left for the conference, he will have more detailed representation of his office and less detailed representation of the conference site. However, when the plane is about to arrive at the destination, his representation of the conference site will be much more detail and he will intentionally use more crude representation for his office.

The above discussion suggests that it might be beneficial for an agent to maintain a non-uniform representation of its knowledge and dynamically adjust the granularity distribution based on context. ln the following, we propose a method of achieving this. We illustrate our approach under the scenario of object search. Our goal is to adjust the representation density based on the target distribution, the higher the density, the more detail the representation.

Our approach is to first obtain the highest granularity map as represented by little cubes. Then, we combine those cubes whose probabilities are not big enough into a larger blob. This process will continue until the probability within the blob is big enough. This effort will result in an mesh representation of the knowledge.

Here is the algorithm:

1. Tessellate the region into small cubes corresponding to the highest granularity.
2. Assign all the small cubes as unconsidered.
3. The process terminates when all the cubes are considered.
4. Find the unconsidered cube with the smallest probability, and mark it as considered.
5. If the probability is smaller than the threshold, then find a neighbor cube that is unconsidered with the smallest probability. Mark this cube as considered. lf there is no such cubes, goto 3.
6. Combine the two together and form the blob. If the probability of the blob is bigger than the threshold, goto 3.
7. Find a neighbor unconsidered cube of the blob with the smallest probability. Mark it as considered and goto 6.

7 Conclusion

The message derived from both the case study and the general analysis is that knowledge granularity has a big impact on the performance of an agent. Thus, an appropriate knowledge granularity should be selected by an agent in order to guarantee a satisfactory result. In complex situations, the selection of granularity depends on many factors. ln general, we can construct graphs like Figure 2 to analyze the effects of granularity on the performance of the agent under different factors and constraints, and then select a favorable granularity. We may also use an adaptive mesh to represent knowledge in some situations.

References

[1] R. Bajcsy. Active perception vs. passive perception. ln *Third IEEE Workshop on Vision*, pages 55–59, Bellaire, 1985.

[2] R. A. Brooks. lntelligence without representation. *Artificial Intelligence*, (47):139–160, 1991.

[3] l. Ferguson. *Touring Machine: An Architecture for Dynamic, Rational, Mobile Agents.* PhD thesis, University of Cambridge, UK, 1992.

[4] T. D. Garvey. Perceptual strategies for purposive vision. Technical Report Technical Note 117, SRl lnternational, 1976.

[5] M. Georgeff and A. Lansky. Reactive reasoning and planning. ln *Proceedings of the Sixth National Conference on Artificial Intelligence*, pages 677–682, Seattle, WA, 1987.

[6] A. Meystel. Theoritical foundations of planning and navigation for autonomous mobile systems. *International Journal of Intelligent Systems*, 2(2):73–128, 1987.

[7] A. Meystel. Multiresolutional architectures for autonomous systems with incomplete and inadequate knowledge representation. In *Artificial Intelligence in Industrial Decision Making, Control, and Automation. Eds. S.G. Tzafestas and H.B. Verbruggen. Kluwer Academic*, pages 159–223, 1995.

[8] A. Meystel, R. Bhatt, D. Gaw, P. Graglia, and S. Waldon. Multiresolutional pyramidal knowledge representation and algorithmic basis of imas-2. ln *Proc. of Mobile Robots, SPIE Vol. 851*, pages 80–116, 1987.

[9] J. Muller and M. Pischel. Modelling interacting agents in dynamic environments. ln *Proceedings of the Eleventh European Conference on Artificial Intelligence*, pages 709–713, Amsterdam, The Netherland, 1994.

[10] S. Sen, S. Roychowdhury, and N. Arora. Effects of local information on group behavior. ln *Proceedings of Second International Conference on Multi-Agent Systems*, pages 315–321, Kyoto, Japan, 1996.

[11] T. Tyrrell. *Computational Mechanisms for Action Selection.* PhD thesis, Center for Cognitive Science, University of Edinburgh, England, 1993.

[12] M. Wooldridge and N. Jenning. lntelligent agents: theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.

[13] Y. Ye. *Sensor Planning for Object Search.* PhD thesis, University of Toronto, Toronto, Canada, January 1997.

[14] Y. Ye and J. K. Tsotsos. Sensor planning in 3d object search: its formulation and complexity. In *The 4th International Symposium on Artificial Intelligence and Mathematics*, Florida, U.S.A., January 3-5 1996.

[15] Y. Ye and J. K. Tsotsos. Knowledge difference and its influence on a search agent. In *First International Conference on AUTONOMOUS AGENTS*, Marina del Rey, CA, January 1997b.

[16] Y. Ye and J. K. Tsotsos. Knowledge granularity and action selection. In *Eighth International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, pages 475–489, 1998.

[17] Y. Ye and J. K. Tsotsos. Sensor planning for 3d object search. *Computer Vision and Image Understanding*, 73(2):145–169, February 1999.

PART II
RESEARCH PAPERS

8. EXPLORATORY ISSUES

FIPER: An Intelligent System for the Optimal Design of Highly Engineered Products

Michael W. Bailey, *GE Aircraft Engines, Cincinnati, OH 45215*
and William H. VerDuin, *OAI, Cleveland, OH 44142*

Abstract

This paper outlines the development of an advanced design environment that invokes a new intelligent system paradigm for the design of highly engineered products. The paradigm of the CAD Master Model (MM) is extended with the introduction of the Intelligent Master Model (IMM). The use of knowledge based engineering tools captures *why* and *how* of the design in addition to the *what.*

Turbine engine development is a highly coupled disciplinary process. With ever increasing demands in life cycle costs, environmental aspects (noise, emissions and fuel consumption) and performance, the availability of accurate analytical tools during the design process is a given and ceases to be a discriminator between competitors. The application of these tools and their automated interaction in a robust computational environment may determine the success or failure of a project by reducing design cycle time and avoiding costly rework.

This paper describes pilot projects at GE Aircraft Engines (GEAE) and the productivity metrics that justified broader implementation within GEAE. Developed using the UniGraphics CAD system for the design of aircraft engines, this system is applicable to any highly engineered product. This approach will, with the support of a four year $21.5M NIST ATP (National Institute of Standards and Technology Advanced Technology Program), be generalized in FIPER (Federated Intelligent Product EnviRonment), a web based environment that will support multi-disciplinary design and optimization.

The Problem

The development *of robust and optimal, highly engineered products and processes* in today's environment of step-function reductions in cycle time, cost take-out, and improved performance seriously tax the capabilities of today's design systems. Further exacerbating the problem is the need to improve and control quality, for both internally manufactured parts and materials and parts produced through supply chains. Since products are now designed, manufactured and serviced at geographically disparate locations, the ability to share relevant product data is critical.

The Solution

FIPER presents a solution in the form of an Integrated Multidisciplinary Design System which

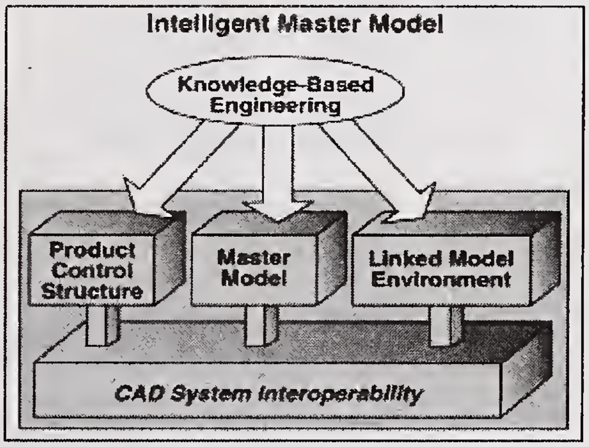
- Exploits the concept of the IMM, permitting context specific views of the MM
- Seamlessly integrates relevant technologies to enable rapid instantiation and simulation-based evaluation of products and processes

Vision: Integrated Multidisciplinary Design Environment

The integrated multidisciplinary design environment under development will enable users to define process maps and rapidly integrate their own proprietary product-specific design and simulation tools through visual programming techniques. It will automatically provide access to a set of technologies including CAD systems and low and high fidelity analysis modules, as well as Multidisciplinary Optimization (MDO) and Robust Design technologies. It will exploit Knowledge Based Engineering to capture rules and best practices that can drive product definition through the (IMM)

Intelligent Master Model

The Intelligent Master Model (Figure 1) is a major enhancement to the Master Modeling concept. Knowledge Based Engineering (KBE) is fused with Product Control Structure (PCS), conventional MM and Linked Model Environment (LME) to collectively render it an Intelligent Master Model. The IMM captures the intent behind the product design by representing the *why* and *how*, in addition to the *what* of a design. The geometric description is only one view of the information associated with the total product model. The IMM can also contain part dependencies, geometric and non-geometric attributes, manufacturing producibility and cost constraints. IMM can provide access to external databases, and can be integrated with proprietary and commercial codes through the LME.


Intelligent Master Model
Knowledge-Based Engineering
Product Control Structure
Master Model
Linked Model Environment
CAD System Interoperability

The IMM can

Figure 1. Intelligent Master Model

capture and archive corporate design practices as well as design and manufacturing engineering expertise. This knowledge can enable less experienced engineers to consistently produce correct first time designs.

The IMM captures the process for generating the PCS at the conceptual and preliminary design level, which then flows the critical information to the detail design and manufacturing. The IMM uses its knowledge base to enable parametric scaling of designs in a top down fashion. When parameters must be computed by execution of simulation codes, the IMM manages this execution by working with process integration tools.

The Master Model

The Master Model captures the requisite information, geometric and non-geometric, to enable context-specific views of necessary design, manufacture, test, and service data. A product design system that supports early requirements definition and flow-down demands that the underlying representation be flexible to geometric, attribute, feature and knowledge-based changes. The traditional CAD representation is flexible only in a geometric sense.

The Master Model (Figure 2) at the lowest or geometric level consists of parametric geometry features such as primitives, extrusions, holes, etc., which form the basic product description. Parameters associated with these geometric features are a subset of the key characteristics which are manipulated to define the product. At this level, the key characteristics include the traditional concepts of dimensionality (length, radius, angle, etc.), as well as those concepts that follow from knowledge-based solid modeling such as offset, spatial alignment, and perpendicularity constraints. Additionally, the existence of a feature is itself an attribute which may be turned on or off as needed to represent the part to varying fidelity levels. For example a bolthole is typically present during a stress analysis but omitted during a computational fluid dynamics analysis. This simplification would be part of the context model, thus creating a context-specific view of the geometry using feature suppression.

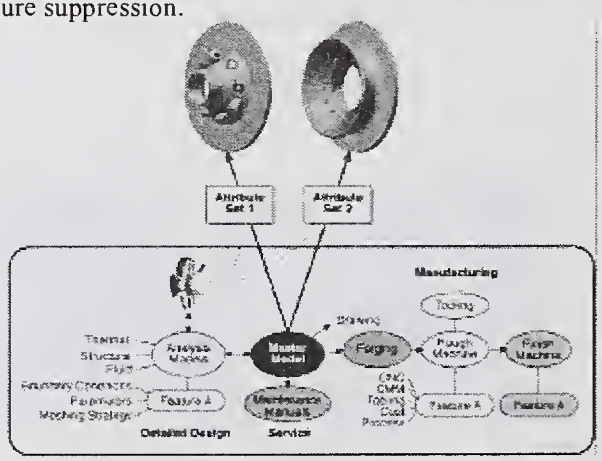


Figure 2. The Master Model supports Feature based Modeling

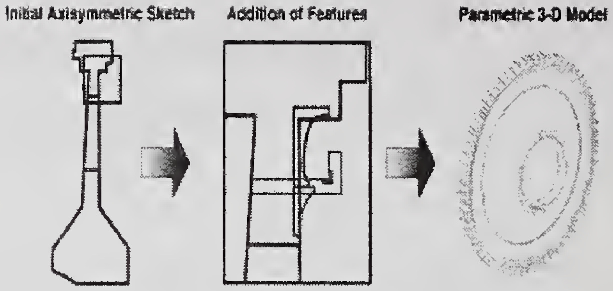


Figure 3. Feature Based Modeling

Using parametric feature-based technology, models are constructed by initially creating simple parametric block shapes to which features (e.g. flanges) are attached. Compound blends are then created and added to the model together with standard features such as radii and chamfers, to create the axisymetric solid. Finally, non-axisymetric features such as holes and slots are then added as shown in Figure 3. This feature-based approach is consistent with feature based analytical model building and cost estimating, while also providing feature suppression functionality.

The initial approach to KBE was the encapsulation of product rules within UniGraphics XESS spreadsheets. These spreadsheets are linked to the geometry such that design rules and practices are parameterized to drive geometry. External codes such as those for disk design could also be executed. Thus an increase in flow thorough the compressor would initiate an aerodynamic resizing of blades and vanes resulting in a blade platform and attachment resizing combined with a disk redesign due to increased centrifugal loads. The whole compressor would thus "rubber band" or parametrically expand to accommodate increased flow.

The Product Control Structure

The PCS facilitates top-down control of the design, allowing the engineer to layout the system configuration and control changes in a top-down fashion. It facilitates *what-if* analysis at the conceptual, preliminary, and detailed design levels by allowing the designer to make parametric changes or to evaluate alternate configurations. This encourages design reuse and enforces standardization in the design process.

The PCS is a hierarchical decomposition of the product into its systems, subsystems and components (Figure 4). These are represented by high-level product attributes and key datum planes and axes to capture their spatial location and orientation. Once the top-level datums have been established and referenced by the subsystems, each subsystem can be designed independently in a distributed manner and later be automatically assembled. Within the PCS, components may be represented by preliminary, simplified geometry (e.g., 2-D cross-sections) or just datums. The cross-sections are picked from a library of cross-section types based on rules. The values for the parameters that define a cross-section are determined using rules captured in the knowledge base. The leaf nodes of the PCS become the seed parts for the bottom-up design of the product into a 3-D assembly. The parts contain 3-D features to capture additional design and manufacturing intent. Everything is fully associative, and thus all changes to the PCS propagate throughout the model.

The Linked Model Environment

Disciplines such as stress analysis, heat transfer analysis, fluids or combustion analysis, and manufacturing and cost prediction each use their own abstraction of the physical model of the product. Within one discipline, several context-specific views may exist as the design evolves. For example, 2-D axisymmetric stress analysis models and detailed 3-D stress analysis models of various levels of refinement for the individual components of a jet engine are required. Each of these analysis models is associated with one or more simulation tools or codes, from simple response surfaces or performance maps during the conceptual design phase, to more complex analysis codes for detailed design, manufacturing process simulation, and cost modeling. This provides the promise of geometric zooming. Historically,

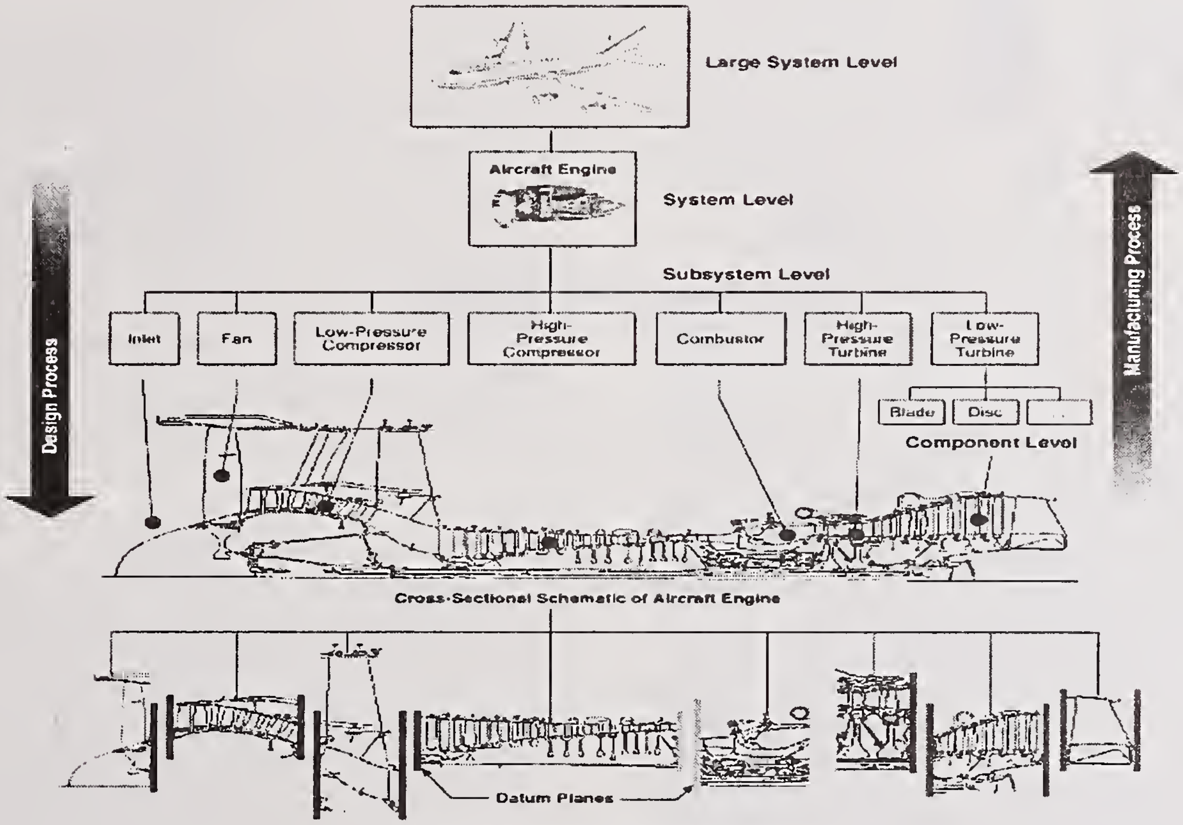


Figure 4. Product Control Structure

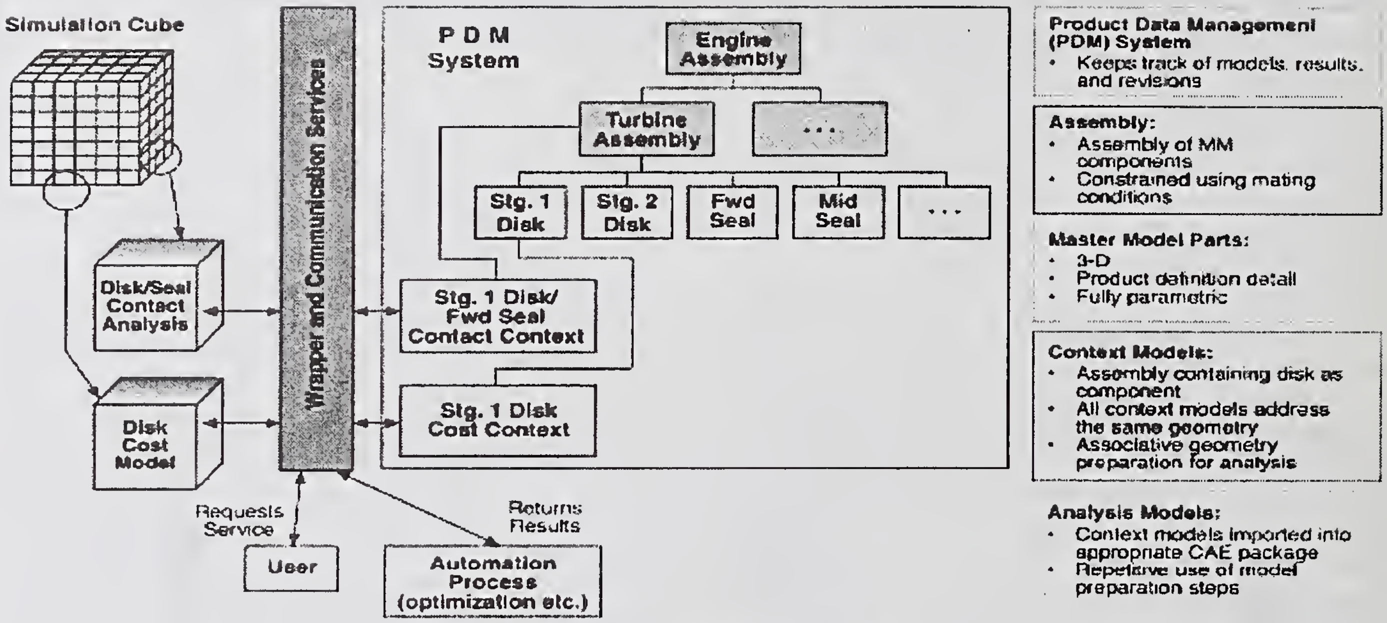

Simulation Cube
Wrapper and Communication Services
P D M System
Engine Assembly
Turbine Assembly
...
Stg. 1 Disk
Stg. 2 Disk
Fwd Seal
Mid Seal
...
Disk/Seal Contact Analysis
Stg. 1 Disk/ Fwd Seal Contact Context
Disk Cost Model
Stg. 1 Disk Cost Context
Requests Service
User
Returns Results
Automation Process (optimization etc.)
Product Data Management (PDM) System
• Keeps track of models, results, and revisions
Assembly:
• Assembly of MM components
• Constrained using mating conditions
Master Model Parts:
• 3-D
• Product definition detail
• Fully parametric
Context Models:
• Assembly containing disk as component
• All context models address the same geometry
• Associative geometry preparation for analysis
Analysis Models:
• Context models imported into appropriate CAE package
• Repetitive use of model preparation steps

Figure 5. Linked Model Environment

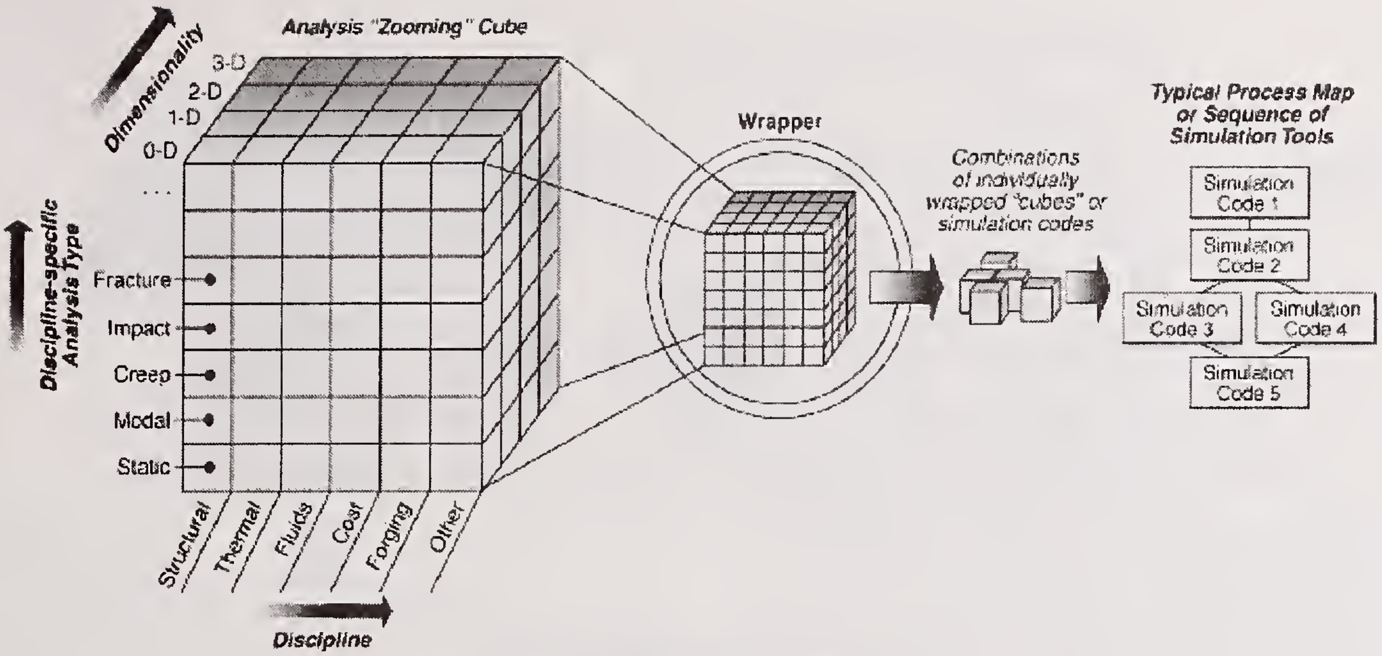


Figure 6. The Simulation Engine

these models exist in a heterogeneous environment, without explicit connections between them. Thus, a design change demanded by one disciplinary group has to be manually incorporated into all the various models of the product that co-exist; a process that is both tedious and error prone. Within the LME (Figure 5) a product's analysis and process models are linked to the Master Model so that all models are automatically synchronized to a single Master Model. Thus, a process is established by which design changes caused by one discipline are fed back to the Master Model. A Product Data Management (PDM) system tracks the design revisions and the associated analysis views or context models of the product.

Simulation Engine

An integral part of the LME is the simulation engine where the analysis tools themselves are wrapped for ease of reuse in a plug-and-pay architecture. To achieve robust and optimal designs, iterative analysis is required. Therefore, ready access to the requisite analysis codes and process maps is essential. The Simulation Engine (Figure 6) provides:

- a programmable mechanism to specify and control the execution of the analysis process
- a mechanism to enable users to easily wrap codes
- an ensemble of pre-wrapped multidisciplinary, variable-fidelity, product-specific analysis tools
- Individual codes and process maps to be linked to the IMM for either manual or automatic execution under program control.

The NASA Glenn Research Center's Numerical Propulsion System Simulator (NPSS) has used a similar cube representation to show the interconnectivity of functional codes, multiple levels of analysis, and zooming to represent their computer-based engine in a test cell. The Simulation Engine is a generalization of this concept for generic products.

Design For Six Sigma

The goal of Design For Six Sigma (DFSS, 3.4 defects per million opportunities) is to create products and processes which are at Six Sigma levels of performance, manufacturability, reliability and cost. DFSS is based on an orderly process which identifies and flows down Critical to Quality (CTQ) characteristics for the product, process or service. This enables quality measures to be driven into the product during the early design phases where the cost of implementing changes is relatively low in comparison to fixing the problems later in the product life cycle. Key design factors for each CTQ are identified and statistical performance models are developed. Modeling, simulation, Design of Experiments (DoE) and analysis are usually employed to develop the statistical models. The essence of DFSS is to migrate from a deterministic to a probabilistic design approach. DFSS is generally focused on shifting means for CTQ's and reducing variances about means so that customer expectations are met at minimum cost.

Robust Design is an intrinsic part of DFSS. Traditionally optimal design and robust design were viewed as independent technologies, but in fact there is great synergism and common core concepts that can be exploited to achieve

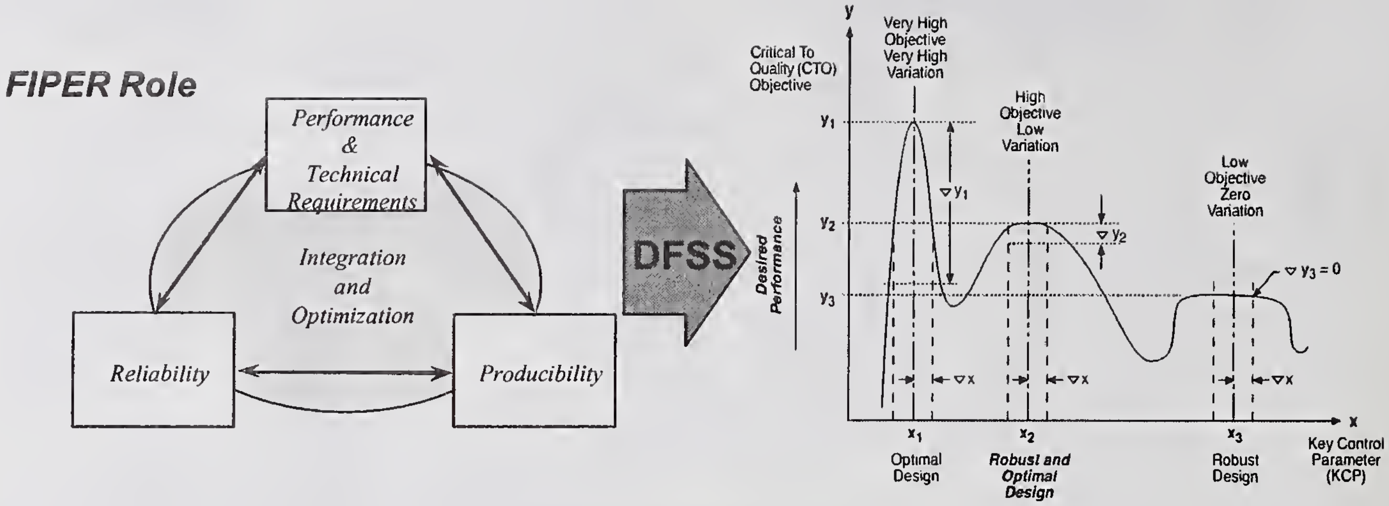

FIPER Role
Performance & Technical Requirements
Integration and Optimization
Reliability
Producibility
DFSS
y
Critical To Quality (CTQ) Objective
Desired Performance
Very High Objective Very High Variation
High Objective Low Variation
Low Objective Zero Variation
y_1
y_2
y_3
∇y_1
∇y_2
$\nabla y_3 = 0$
∇x
x
x_1
Optimal Design
x_2
Robust and Optimal Design
x_3
Robust Design
Key Control Parameter (KCP)

Figure 7. The True Meaning of DFSS

optimal *and* robust designs for products and processes. Optimality and robustness often have competing objectives. The focus of the robust optimization problem is to simultaneously optimize the performance (mean of the response) and minimize the variation. In other words, a maximization problem would not merely strive for the highest peak, but would strive for a high plateau. *In practice this represents a trade-off between Performance and Technical Requirements, Reliability and Producibility* (Figure 6). This represents a paradigm shift in design methodology.

Background

There are many definitions of a Master Model. At GEAE the definition is a single geometric representation, ideally 3-D, created at concept using feature based parametric modeling techniques in a linked associative environment, and utilized through manufacturing. In addition there is an evolution of a tight integration of all elements of a product creation, manufacturing and support permitting true concurrency for analysis and manufacturing since updates can be flowed down to the individual activities from the MM. An additional requirement is the management of all types of data or metadata within the Common Geometry environment. The fusion of a conventional MM with PCS, LME and KBE results in an IMM, the next logical step in CAD's evolution.

Historically analysis codes were coupled together with input and output files; geometry was provided as an output as necessary, probably as an IGES file. The new approach is to have geometry central or *common* to all processes and to use it as a design integrator. This facilitates CAD integration with analysis and manufacturing. Four years ago GE Aircraft Engines started its Common Geometry initiative, based on UniGraphics and commercial code to the extent possible. The first year focused on strategy. Historically at GEAE conceptual and preliminary design are accomplished using simplifying assumptions in a unique set of tools. Changes in the underlying assumptions and the lack of a rigorous handoff to detail design often meant that the preliminary design was repeated. Since business commitments are made based on preliminary design this increased the risk of meeting customer CTQ requirements. It is well understood that 70 to 80% of a product's cost is locked in during conceptual and preliminary design. Previous efforts had focused on productivity tools that relied heavily on automation. The discovery of UG/WAVE with its top down approach using a Control Structure meant it was possible to drive the design using requirements providing *functional* and *spatial* integration thereby making it possible to create 3-D solid models at the Conceptual/Preliminary Design Phase. This combined with a tight integration of CAD with analysis and manufacturing in LME would provide a truly concurrent design environment.

During the second year three pilots were conducted to demonstrate the technical feasibility and generate metrics for the return on investment analysis necessary to move to a broader implementation across the business. These pilots focused on Conceptual/Preliminary design, Detailed design and Manufacturing. Although these pilots addressed different sections of the engine, success in individual areas would provide confidence to proceed to a broader implementation.

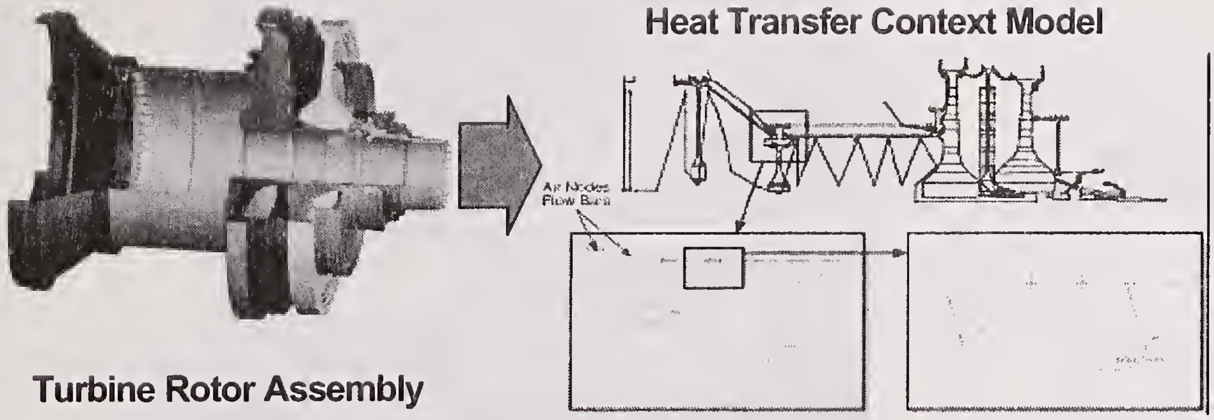


Figure 8. LME Engineering Pilot

SOLID Pilot

The purpose of the SOLID (System Oriented Layout with Integrated Design) pilot was to build a 3-D solid geometry model of a compressor. This was constructed using the UG/WAVE PCS. Model construction is of paramount importance if productivity gains downstream are to be realized. Constructing the model from features enables suppression of selective features by downstream users using context models or "views of geometry". Traditional 2-D axisymetric cross sections can still be generated from the 3-D solid. These would be completely associative to the solid and would constitute an output instead of an input. Thus the parameters that drive the 3-D solid would also drive the 2-D cross section. Time invested in constructing the 3-D models facilitates updates as the design evolves. By segregating out the work that would be eliminated using the SOLID model from the charging data from a recently completed program, it was estimated that 34% would be saved at the Conceptual/Preliminary design phase and 7% at the Detailed Design phase.

A key element in the Integration of CAD with Analysis, or any geometry dependent activity, is the creation of context models. A Context model uses the concept of CAD Assemblies to create a "view" of geometry. Just as it is conventional CAD practice to combine parts into assemblies building up into the complete system, it is possible to combine geometry with context information in the form of an assembly. Context in this application means the attachment of information necessary to create a structural, thermal or Computational Fluid Dynamics (CFD) model to geometric entities. The rotor assembly could also be regarded as a context model. This information could be boundary conditions such as pressures, temperatures, loads and the meshing strategy such as mesh seeds or mesh densities. These attributes are applied to the geometric entities in the CAD package.

This context information or "Tagging" should be robust to parametric or non-topological changes and have some robustness to topological changes. A longer term goal is to apply these "Tags" as the analysis model is built in the meshing software, then export these to the CAD software for storage. Currently they are applied in the CAD software. The CAD assembly context model is imported into the meshing software such as PATRAN, ANSYS or ICEM CFD to create the application model. The heat transfer context model is shown in Figure 8. From data accumulated during the pilot, it was estimated that savings of 25% in Detailed Design were possible.

In the manufacturing pilot the focus was using manufacturing context models in conjunction with the 3-D Master Model to generate in process planning and shapes, tooling and Computer Numerically Controlled (CNC) machining tapes. A Low Pressure turbine disk currently in production was used. Note that in the manufacturing environment the modeling works in the opposite sense to detail design. In the detailed design features are added to the model as the design progresses from conceptual through preliminary and detail design; in manufacturing features are removed consistent with manufacturing operations until the raw material remains. Figure 9 shows the associated in-process models and tooling together with the engineering analysis, results and drawing creation. The pilot demonstrated a 15% reduction in process development time and an 80% reduction in process regeneration for parametric changes. In addition the associated tooling was updated when the model was changed.

Computer Measuring Machine (CMM) inspection programs can also be generated from the process models. This is another key context model use of the linked associative environment. Aircraft engine manufacture involves the machining of complex shapes from high temperature alloys that "move" during the manufacturing process. Thus it is important for process control to inspect the process shapes to know what the dimensions are so adjustments can be made to future machining operations. This offers the possibility of a "real time" machining feedback loop.

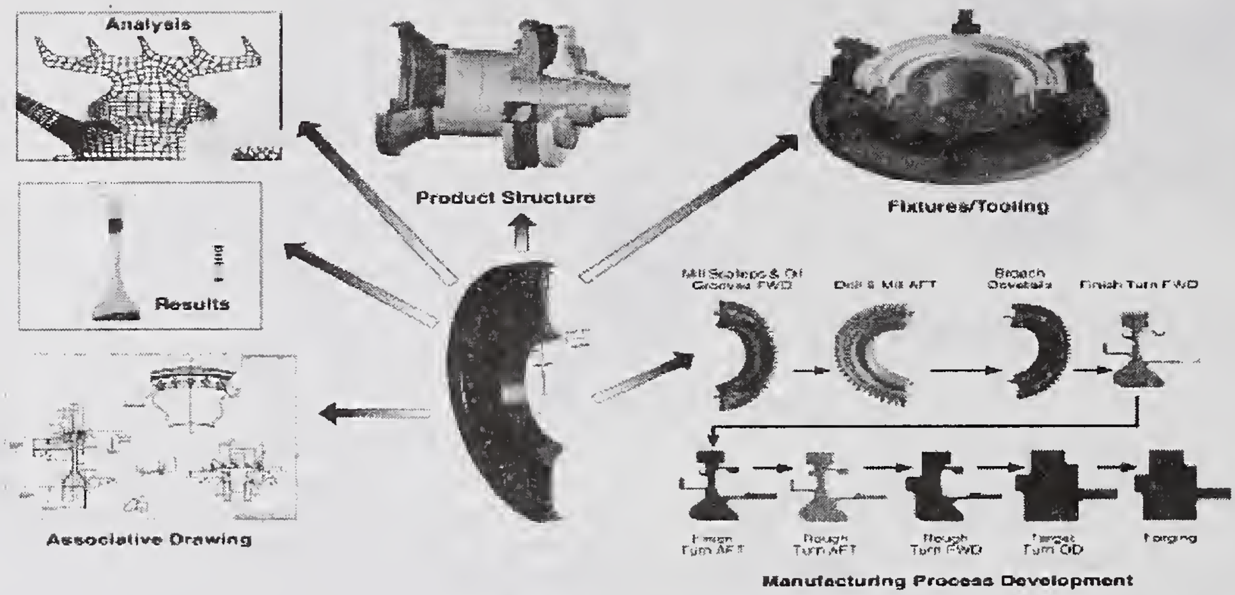


Figure 9. LME Manufacturing Pilot

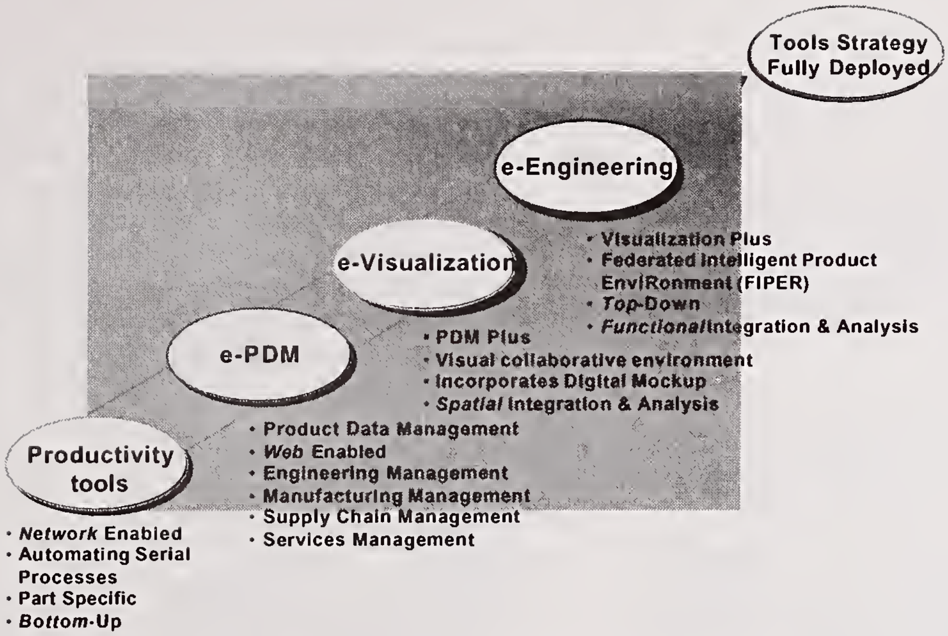

Tools Strategy Fully Deployed
e-Engineering
• Visualization Plus
• Federated Intelligent Product EnviRonment (FIPER)
• Top-Down
• FunctionalIntegration & Analysis
e-Visualization
• PDM Plus
• Visual collaborative environment
• Incorporates Digital Mockup
• Spatial Integration & Analysis
e-PDM
• Product Data Management
• Web Enabled
• Engineering Management
• Manufacturing Management
• Supply Chain Management
• Services Management
Productivity tools
• Network Enabled
• Automating Serial Processes
• Part Specific
• Bottom-Up

Figure 10. Incremental Approach to Development and Deployment

Incremental Approach to Development and Deployment

GEAE's incremental approach to development and deployment is shown in Figure 10.

Productivity Tools/Common Geometry was *network* enabled and automated serial tasks such as mesh creation on individual parts. This can be described as the "run faster" approach and is sub-optimal since it optimizes individual parts in bottom-up design as opposed to the system design.

e-PDM focuses on Product Data Management (PDM) and is W*eb* enabled. PDM typically provides Engineering Management, Manufacturing Management, Supply Chain Management and Services Management with databases, parts lists, process flow, etc. *It would provide the infrastructure for subsequent development.*

e-Visualization represents an enhancement of e-PDM in that it provides a visual collaborative environment incorporating a digital mockup. It thus provides a visual representation of the engineering assembly permitting *spatial* integration and is with functionality such as interference clearance and removal envelope assessment.

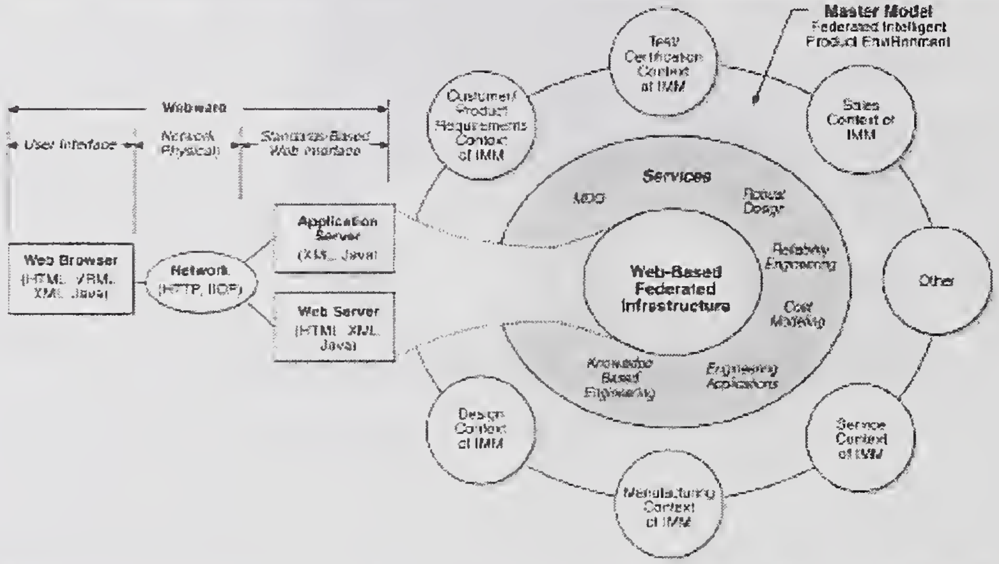

Master Model
Federated Intelligent Product EnviRonment
Webware
User Interface
Network Physical
Standards-Based Web Interface
Web Browser (HTML, VRML, XML, Java)
Network (HTTP, IIOP)
Application Server (XML, Java)
Web Server (HTML, XML, Java)
Test/Certification Context of IMM
Customer/Product Requirements Context of IMM
Sales Context of IMM
Other
Service Context of IMM
Manufacturing Context of IMM
Design Context of IMM
Services
MDO
Robust Design
Reliability Engineering
Cost Modeling
Engineering Applications
Knowledge Based Engineering
Web-Based Federated Infrastructure

Figure 11. Federated Integrated Design EnviRonment

e-Engineering builds on the benefits of e-PDM and e-Visualization to provide an environment that supports *functional* integration and analysis providing a top-down or "run smarter" design environment.

The recent FIPER project award by NIST ATP will provide such an environment. Drawing on the experience and qualifications of the FIPER team members and leveraging GE's Corporate commitment to Design For Six Sigma methodology and products, the proposed program will result in the development, demonstration and transition of advanced tools and technology. Key elements of the NIST ATP include:

- Development of an extensible, standards-based plug and play, Web-based architecture to enable the creation of Six Sigma products and processes.
- Development and major enhancement of a set of advanced core technologies necessary to realize Design For Six Sigma, most notably Intelligent Master Modeling, Knowledge Based Engineering, Robust Design, Multidisciplinary Design and Optimization, Cost Modeling and Producibility.
- Demonstration of FIPER on a diverse set of demanding applications, which span conceptual design, through manufacturing for systems, subsystems and components.
- Dissemination of the technology through a well founded commercialization plan, complimentary teaming, Web-based access, publications, educational programs and the creation of an early adoption program.

Thus FIPER represents a paradigm shift for product development through the introduction of a standards based product development environment Conceptually the FIPER environment is described in Figure 11 and in more detail in Reference1.

The team was chosen for their complimentary roles in achieving the overall FIPER objectives. GEAE is a complex engineering system developer and manufacturer and a Unigraphics CAD system user. Parker Hannifin is a complex aircraft engine and aircraft subsystem and component supplier and a ProEngineer CAD system user. BFGoodrichAerospace is a complex aircraft sub-system and component supplier and CATIA CAD system user. Thus with CAD interoperability being one of the major FIPER initiatives, three out of the four major CAD systems is represented. The fourth, SDRC IDEAS Master Series will be addressed at a later stage, possibly through the early adopter program. GE Corporate Research and Development (CR&D) has been developing the technology associated with IMM, KBE, MDO and DFSS for a number of years. Engineous Software Inc. is the commercializer for the FIPER software and their current product is iSIGHT, an engineering analysis process integration and optimization tool. Ohio University is providing computer system integration software wrapping tools and is developing a cost model that will be integrated with the IMM. Stanford University is creating producibility models that will be integrated with the IMM. OAI (Ohio Aerospace Institute) is the sponsoring organization and provides program administration. The complimentary teaming are key to the technical and commercial success of the FIPER project.

Acknowledgements

This research is jointly funded through the National Institute for Standards and Technology Advanced Technology Program (NIST ATP) and the General Electric Company. The authors would like to acknowledge this support as well as the valuable input from the whole FIPER team.

References

Ref. I: Röhl, P. J., Kolonay, R. M. et al. *A Federated Intelligent Product EnviRonment* AIAA-2000-4902, 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, CA, September 6-8, 2000

Towards an Objective Comparison of Stochastic Optimization Approaches[*]

James C. Spall, Stacy D. Hill, and David R. Stark

The Johns Hopkins University
Applied Physics Laboratory
11100 Johns Hopkins Road
Laurel, Maryland 20723-6099 U.S.A.

ABSTRACT

This paper is a first step to formal comparisons of several leading optimization algorithms, establishing guidance to practitioners for when to use or not use a particular method. The focus in this paper is four general algorithm forms: random search, simultaneous perturbation stochastic approximation, simulated annealing, and evolutionary computation. We summarize the available theoretical results on rates of convergence for the four algorithm forms and then use the theoretical results to draw some preliminary conclusions on the relative efficiency. Our aim is to sort out some of the competing claims of efficiency and to suggest a structure for comparison that is more general and transferable than the usual problem-specific numerical studies. Work remains to be done to generalize and extend the results to problems and algorithms of the type frequently seen in practice.

KEYWORDS: *Rate of convergence; random search; simultaneous perturbation stochastic approximation; simulated annealing; evolutionary computation.*

1. INTRODUCTION

To address the shortcomings of classical deterministic algorithms, a number of powerful optimization algorithms with embedded randomness have been developed. The population-based methods of evolutionary computation are only one class among many of these available *stochastic* optimization algorithms. Hence, a user facing a challenging optimization problem for which a stochastic optimization method is appropriate meets the daunting task of determining which algorithm is appropriate for a given problem. This choice is made more difficult by the large amount of "hype" and dubious claims that are associated with some popular algorithms. An inappropriate approach may lead to a large waste of resources, both from the view of wasted efforts in implementation and from the view of the resulting suboptimal solution to the optimization problem of interest.
Hence, there is a need for objective analysis of the relative merits and shortcomings of leading approaches to stochastic optimization. This need has certainly been recognized by others, as illustrated in the recent 1998 IEEE International Conference on Evolutionary Computation, where one of the major subject divisions in the conference was devoted to comparing algorithms. Nevertheless, virtually all comparisons have been numerical tests on specific problems. Although sometimes enlightening, such comparisons are severely limited in the *general* insight they provide. On the other end of the spectrum are the "No Free Lunch Theorems" (Wolpert and McReady, 1997), which simultaneously considers all possible loss functions and thereby draw conclusions that have limited practical utility since one always has at least *some* knowledge of the nature of the loss function being minimized.

Our aim in this paper is to lay a framework for a *theoretical* comparison of efficiency applicable to a broad class of practical problems where some (incomplete) knowledge is available about the nature of the loss function. We will consider four basic algorithm forms—random search, simultaneous perturbation stochastic approximation (SPSA), simulated annealing, and evolutionary computation via genetic algorithms—in the context of continuous variable optimization. The basic optimization problem corresponds to finding an optimal point θ^*:

$$\theta^* = \arg\min_{\theta \in D} L(\theta),$$

where $L(\theta)$ is the loss function to be minimized, D is the domain over which the search will occur, and θ is a p-

[*] A more complete version of this manuscript is available upon request (james.spall@jhuapl.edu). This work was partially supported by the JHU/APL IRAD Program and U.S. Navy contract N00024-98-D-8124.

dimensional (say) vector of parameters. We are mainly interested in the typical case where θ^* is a *unique* global minimum.

Although many stochastic optimization algorithms other than the four above exist, we are restricting ourselves to the four general forms in order to be able to make tangible progress (note that there are various specific implementations of each of these general algorithm forms). These four algorithms are general-purpose optimizers with powerful capabilities for serious multivariate optimization problems. Further, they have in common the requirement that they only need measurements of the objective function, not requiring the gradient or Hessian of the loss function.

2. NO FREE LUNCH THEOREMS AND THEIR RELATIONSHIP TO RATE OF CONVERGENCE

Wolpert and Macready (1997) present a formal analysis of search algorithms for optimization, the most popular of which are evolutionary computation, simulated annealing (SAN) and random search. This work results in several “No Free Lunch Theorems,” stating, in essence, that no algorithm is universally better than other algorithms. The full version of this paper goes into some detail on the implications of these theorems.

3. SIMPLE GLOBAL RANDOM SEARCH

We first establish a rate of convergence result for the simplest random search method where we repeatedly sample over the domain of interest, $D \subseteq R^p$. This can be done in recursive form or in “batch” (non-recursive) form by simply laying down a number of points in D and taking as our estimate of θ^* that value of θ yielding the lowest L value. It is well known that the random search algorithm above will converge in some stochastic sense under modest conditions (e.g., Solis and Wets, 1981; Spall, 2000b):

To evaluate the *rate* of convergence, let us specify a “satisfactory region” $S(\theta^*)$ representing some neighborhood of θ^* providing acceptable accuracy in our solution (e.g., $S(\theta^*)$ might represent a hypercube about θ^* with the length of each side representing a tolerable error in each coordinate of θ). An expression related to the rate of convergence of Algorithm A is then given by

$$P(\hat{\theta}_k \in S(\theta^*)) = 1 - [1 - P(\theta_{\text{new}}(k) \in S(\theta^*)]^k \qquad (3.1)$$

We will use this expression in Section 7 to derive a convenient formula for comparison of efficiency with other algorithms.

4. SIMULTANEOUS PERTURBATION STOCHASTIC APPROXIMATION

The next algorithm we consider is SPSA. This algorithm is designed for continuous variable optimization problems. Unlike the other algorithms here, SPSA is fundamentally oriented to the case of *noisy* function measurements and most of the theory is in that framework. This will make for a difficult comparison with the other algorithms, but Section 7 will attempt a comparison nonetheless. The SPSA algorithm works by iterating from an initial guess of the optimal θ, where the iteration process depends on a highly efficient "simultaneous perturbation" approximation to the gradient $g(\theta) \equiv \partial L(\theta)/\partial\theta$.

The SPSA procedure is in the general recursive SA form:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \hat{g}_k(\hat{\theta}_k) \qquad (4.1)$$

where $\hat{g}_k(\hat{\theta}_k)$ is the SP estimate of the gradient $g(\theta) \equiv \partial L/\partial\theta$ at the iterate $\hat{\theta}_k$ (Spall, 1992) based on the measurements of the loss function and $a_k > 0$ is a “gain” sequence. This iterate can be shown to converge under reasonable conditions (e.g., Spall, 1992; Dippon and Renz, 1997). The essential basis for efficiency of SPSA in multivariate problems is due to the gradient approximation, which uses only two measurements of the loss function to estimate the p-dimensional gradient vector for any p. This contrasts with the standard finite difference method of gradient approximation, which requires $2p$ measurements.

Most relevant to the comparative analysis goals of this paper is the asymptotic distribution of the iterate. This was derived in Spall (1992), with further developments in Chin (1997), Dippon and Renz (1997), and Spall (2000a). Essentially, it is known that under appropriate conditions,

$$k^{\beta/2}(\hat{\theta}_k - \theta^*) \xrightarrow{\text{dist}} N(\mu, \Sigma) \text{ as } k \to \infty , \qquad (4.2)$$

where $\beta > 0$ depends on the choice of gain sequences (a_k and c_k), μ depends on both the Hessian and the third derivatives of $L(\theta)$ at θ^* (note that in general, $\mu \neq 0$ in contrast to many well-known asymptotic normality results in estimation), and Σ depends on the Hessian matrix at θ^* and the variance of the noise in the loss measurements. Unfortunately, (4.2) is not directly usable in our comparative studies here since the other three algorithms being considered here appear to have convergence rate results only for the case of *noise-free* loss measurements. Recent results by Gerencsér (1999) and Gerencsér and Vágó (2000) on noise-free SPSA may ultimately be useful.

5. SIMULATED ANNEALING ALGORITHMS

The simulated annealing (SAN) method (Metropolis et al., 1953; Kirkpatrick et al., 1983) was originally developed for optimization over finite sets. The Metropolis method produces a sequence that converges in probability to the set of global minima of the loss function as T_k , the *temperature*, converges to zero. Geman and Hwang (1986) present a SAN algorithm for continuous parameter optimization. Their algorithm produces a *continuous-time* stochastic process—a diffusion process—whose probability distributions converge weakly to the uniform probability distribution concentrated on the (global) minima of the loss function, as the temperature decreases to zero.

More recently, Gelfand and Mitter (1993) obtained discrete-time recursions for Metropolis-type SAN algorithms that, in the limit, optimize continuous parameter loss functions: Suppose that $\{\hat{\theta}_k\}$ is a Metropolis SAN sequence for optimizing L and assume that the gradient g of L exists (it does not have to be actually computed).

Furthermore, like SPSA, SAN has an asymptotic normality result (but unlike SPSA, this result applies in the noise-free case). Let $H(\theta^*)$ denote the Hessian of $L(\theta)$ evaluated at θ^* and let I_p denote the $p \times p$ identity matrix. Yin (1999) showed that for $b_k = (b/(k^\gamma \log(k^{1-\gamma} + B_0))^{1/2}$,

$$[\log(k^{1-\gamma} + B_0)]^{1/2}(\hat{\theta}_k - \theta^*) \rightarrow N(0, \Sigma) \text{ in distribution,}$$

where $\Sigma H + H^T\Sigma + (b/a)I = 0$.

6. EVOLUTIONARY COMPUTATION

There are three general approaches in evolutionary computation, namely Evolutionary Programming (EP), Evolutionary Strategies (ES) and Genetic Algorithms (GA). All three approaches work with a population of candidate solutions and randomly alter the solutions over a sequence of generations according to evolutionary operations of competitive selection, mutation and sometimes recombination (reproduction). The fitness of each population element to survive into the next generation is determined by a selection scheme based on evaluating the loss function for each element of the population. The selection scheme is such that the most favorable elements of the population tend to survive into the next generation while the unfavorable elements tend to perish.

The principle differences in the three approaches are the selection of evolutionary operators used to perform the search and the computer representation of the candidate solutions. EP uses selection and mutation only to generate new solutions. While both ES and GA use selection, recombination and mutation, recombination is used more extensively in GA. A GA traditionally performs evolutionary operations using binary encoding of the solution space, while EP and ES perform the operations using real-coded solutions. The GA also has a real-coded form and there is some indication that the real-coded GA may be more efficient and provide greater precision than the binary-coded GA. The distinction among the three approaches has begun to blur as new hybrid versions of EC algorithms have arisen.

Global convergence results can be given for a broad class of problems, but the same can not be said for convergence *rates*. The most practically useful convergence rates for EC algorithms seem to be for the class of strongly convex fitness functions. The following result due to Rudolph (1997b) is an extension of a more general result by Rappl (1989). The theorem will be the starting place for the specific convergence rate result that will be used for comparison in Section 7. A more complete discussion of the relevant EC theory is in the full version of the paper.

An EC algorithm has a *geometric rate of convergence* if and only if $E[L_k^* - L(\theta^*)] = O(c^k)$ where $c \in (0, 1)$ is called the *convergence rate*. Under conditions, the convergence rate result for a (1, λ)-ES using selection and mutation only on a strongly convex fitness function is geometric with a rate of convergence

$$c = (1 - M_{\lambda,p}^2/Q^2) \text{ where } M_{\lambda,p} = E[B_{\lambda:\lambda}]>0$$

and where $B_{\lambda:\lambda}$ denotes the maximum of λ independent identically distributed Beta random variables. The computation of $M_{\lambda,p}$ is apparently very complicated since it depends on both the number of offspring λ and the problem dimension p. An asymptotic approximation for the convergence rate for the (N, λ)-ES where offspring are only obtained by mutation is $c \leq [1 - (2p^{-1}\log(\lambda/N))/Q^2]$.

7. COMPARATIVE ANALYSIS

7.1 Problem Statement and Summary of Efficiency Theory for the Four Algorithms

This section uses the specific algorithm results in Sections 3 to 6 above in drawing conclusions on the relative performance of the four algorithms. There are obviously many ways one can express the rate of convergence, but it is expected that, to the extent they are based on the theory outlined above, the various ways will lead to broadly similar conclusions. We will address the rate of convergence by focusing on the question:

With some high probability $1-\rho$ *(*ρ *a small number), how many* $L(\cdot)$ *function evaluations, say n, are needed to achieve a solution lying in some "satisfactory set"* $S(\theta^*)$ *containing* θ^**?*

For each of the four algorithms, we will outline below an analytical expression useful in addressing the question. After we have discussed the analytical expressions, we present a comparative analysis in a simple problem setting for varying *p*.

Random Search

We can use (3.1) to answer the question above. Setting the left-hand side of (3.2) to $1-\rho$ and supposing that there is a constant sampling probability $P^* = P(\theta_{\text{new}}(k) \in S(\theta^*))\ \forall\ k$, we have

$$n = \frac{\log\rho}{\log(1-P^*)}. \qquad (7.1)$$

Simultaneous Perturbation Stochastic Approximation

From the fact that SPSA uses two $L(\theta^*)$ evaluations per iteration, the value *n* to achieve the desired probability for $\hat{\theta}_k \in S(\theta^*)$ is then

$$n = 2\left(\frac{2d(p)\sigma}{\delta s}\right)^3.$$

Simulated Annealing

The value *n* to achieve the desired probability for $\hat{\theta}_k \in S(\theta^*)$ is

$$\log n^{1-\gamma} = \left(\frac{2d(p)\sigma}{\delta s}\right)^2.$$

Evolutionary Strategy

The full version of the paper employs Markov's inequality and the bound in Rudolph (1997b) to show that for each generation *k*, there are λ evaluations of the fitness function so that $n = \lambda k$, where

$$k = \frac{\log\rho - \log(1/\varepsilon)}{\log\left[1 - \frac{2}{pQ^2}\log(\lambda/N)\right]}.$$

7.2 Application of Convergence Rate Expressions for Varying p

We now apply the results above to demonstrate relative efficiency for varying *p*. Let $D = [0, 1]^p$ (the *p*-dimensional hypercube with minimum and maximum θ values of 0 and 1 for each component). We want to guarantee with probability 0.90 that each element of θ is within 0.04 units of the optimal. Let the (unknown) true θ, θ^*, lie in $(0.04, 0.96)^p$. The individual components of θ^* are θ_i^*. Hence,

$$S(\theta^*) = [\theta_1^* - 0.04, \theta_1^* + 0.04] \times [\theta_2^* - 0.04, \theta_2^* + 0.04] \times \ldots \times [\theta_p^* - 0.04, \theta_p^* + 0.04] \subset D.$$

Table 7.1 is a summary of relative efficiency for the setting above for $p = 2$, 5, and 10; the efficiency was normalized so that all algorithms performed equally at $p = 1$, as described below. The numbers in Table 7.1 are the ratios of the number of loss measurements for the given algorithm over the number for the best algorithm at the specified *p*; the highlighted values 1.0 indicate the best algorithm for each of the values of *p*. To establish a fair basis for comparison, we fixed the various parameters in the expressions above (e.g., σ in SPSA and SAN, ρ for the ES, etc.) so that the algorithms produced identical efficiency results for $p = 1$.

Table 7.1. Ratios of loss measurements needed relative to best algorithm at each p for $1 \leq p \leq 10$

	$p = 1$	$p = 2$	$p = 5$	$p = 10$
Rand. Search	$\boxed{1.0}$	11.6	8970	2.0×10^9
SPSA	$\boxed{1.0}$	1.5	$\boxed{1.0}$	$\boxed{1.0}$
SAN	$\boxed{1.0}$	$\boxed{1.0}$	2.2	4.1
ES	$\boxed{1.0}$	1.9	1.9	2.8

Table 7.1 illustrates the explosive growth in the relative (and absolute) number of loss evaluations needed as p increases for the random search algorithm. The other algorithms perform more comparably, but there are still some non-negligible differences. For example, at $p = 5$, SAN will take 2.2 times more loss measurements than SPSA to achieve the objective of having $\hat{\theta}_k$ inside $S(\theta^*)$ with probability 0.90. Of course, as p increases, all algorithms take more measurements; the table only shows relative numbers of function evaluations (considered more reliable than absolute numbers).

This large improvement of SPSA and SAN relative to random search may partly result from the more restrictive regularity conditions of SPSA and SAN (i.e., for formal convergence, SPSA assumes a unimodal, several-times-differentiable loss function) and partly from the fact that SPSA and SAN work with *implicit* gradient information via gradient approximations. The performance for ES is quite good. The restriction to strongly convex fitness functions, however, gives the ES in this setting a strong structure not available to the other algorithms. It remains unclear what practical theoretical conclusions can be drawn on a broader class of problems.

REFERENCES

Bäck, T., Hoffmeister, F., and Schwefel, H.-P. (1991), "A Survey of Evolution Strategies," in *Proceedings of the Fourth International Conference on Genetic Algorithms* (R.K. Belew and L.B. Booker, eds.), pp. 2-9.

Beyer, H.-G. (1995), "Toward a Theory of Evolution Strategies: On the Benefits of Sex—the ($\mu/\mu,\lambda$) Theory," *Evolutionary Computation,* vol. 3, pp. 81-111.

Chin, D.C. (1994), "A More Efficient Global Optimization Algorithm Based on Styblinski and Tang," *Neural Networks*, vol. 7, pp. 573-574.

Chin, D.C. (1997), "Comparative Study of Stochastic Algorithms for System Optimization Based On Gradient Approximations," *IEEE Transactions on Systems, Man, and Cybernetics—B*, vol. 27, pp. 244-249.

Culberson, J.C. (1998), "On the Futility of Blind Search: An Algorithmic View of 'No Free Lunch'," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 109-127.

Dippon, J. and Renz, J. (1997), "Weighted Means in Stochastic Approximation of Minima," *SIAM Journal on Control and Optimization*, vol. 35, pp. 1811-1827.

Fabian, V. (1968), "On Asymptotic Normality in Stochastic Approximation," *Annals of Mathematical Statistics*, vol. 39, pp. 1327-1332.

Gelfand, S. and Mitter, S.K. (1993), "Metropolis-Type Annealing Algorithms for Global Optimization in R^d," *SIAM Journal of Control and Optimization*, vol. 31, pp. 111-131.

Geman, S. and Hwang, C.-R. (1986), "Diffusions for Global Optimization," *SIAM Journal of Control and Optimization*, vol. 24, pp. 1031-1043.

Gerencsér, L. (1999), "Convergence Rate of Moments in Stochastic Approximation with Simultaneous Perturbation Gradient Approximation and Resetting," *IEEE Transactions on Automatic Control*, vol. 44, pp. 894-905.

Gerencsér, L. and Vágó, Z. (2000), "SPSA in Noise-Free Optimization," in *Proceedings of the American Control Conference*, pp. 3284-3288.

Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P. (1983), "Optimization by Simulated Annealing," *Science*, vol. 220, pp. 671-680.

Maryak, J.L. and Chin, D.C. (2000), "Stochastic Approximation for Global Random Optimization," in *Proceedings of the American Control Conference*, pp. 3294-3298.

Metropolis, N., Rosenbluth, A., Rosenbluth, M. Teller, A. and Teller, E. (1953), "Equation of State Calculations by Fast Computing Machines," *Journal of Chemical Physics*, vol. 21, pp. 1087-1092.

Nemirovsky, A.S. and Yudin, D.B (1983), *Problem Complexity and Method Efficiency in Optimization*, Wiley, Chichester.

Rappl, G. (1989), "On Linear Convergence of a Class of Random Search Algorithms", *Zeitschrift für angewandt Mathematik und Mechanik (ZAMM)*, vol. 69, pp. 37-45.

Rudolph, G. (1994), "Convergence Analysis of Canonical Genetic Algorithms", *IEEE Transactions on Neural Networks*, vol. 5, no. 1, pp. 96-101.

Rudolph, G. (1996), "Convergence of Evolutionary Algorithms in General Search Spaces," in *Proceedings of the Third IEEE Conference on Evolutionary Computation*, pp. 50-54.

Rudolph, G. (1997a), *Convergence Properties of Evolutionary Algorithms,* Kovac, Hamburg

Rudolph, G. (1997b), "Convergence Rates of Evolutionary Algorithms for a Class of Convex Objective Functions," *Control and Cybernetics*, vol. 26, pp. 375-390.

Rudolph, G. (1998), "Finite Markov Chain Results in Evolutionary Computation: A Tour d'Horizon," *Fundamenta Informaticae*, vol. 34, pp. 1-22.

Solis, F.J. and Wets, J.B. (1981), "Minimization by Random Search Techniques," *Mathematics of Operations Research*, vol. 6, pp. 19-30.

Spall, J.C. (1992), "Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation," *IEEE Transactions on Automatic Control*, vol. 37, pp. 332–341.

Spall, J.C. (2000a), "Adaptive Stochastic Approximation by the Simultaneous Perturbation Method," *IEEE Transactions on Automatic Control*, vol. 45, in press.

Spall, J.C. (2000b), *Introduction to Stochastic Search and Optimization*, Wiley, New York, in preparation.

Tong, Y.L. (1980), *Probability Inequalities in Multivariate Distributions,* Academic, New York.

Wolpert. D.H. and Macready, W.G. (1997), "No Free Lunch Theorems for Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, pp. 67-82.

Yin, G.G. (1999), "Rates of Convergence for a Class of Global Stochastic Optimization Algorithms," *SIAM Journal on Optimization*, vol. 10, pp. 99-120.

Some Measurable Characteristics of Intelligent Computing Systems

Christopher Landauer, Kirstie Bellman
Aerospace Integration Science Center
The Aerospace Corporation, Mail Stop M6/214
P. O. Box 92957, Los Angeles, California 90009-2957, USA
cal@aero.org, bellman@aero.org

Abstract

We discuss the following measurable characteristics of intelligent behavior in computing systems: (1) speed and scope of adaptibility to unforeseen situations, including recognition, assessment, proposals, selection, and execution; (2) rate of effective learning of observations, behavior patterns, facts, tools, methods, etc., which requires identification, encapsulation, and recall; (3) accurate modeling and prediction of the relevant external environment, which includes the ability to make more effective abstractions; (4) speed and clarity of problem identification and formulation; (5) effective association and evaluation of disparate information; (6) identification of more important assumptions and prerequisites; (7) use of symbolic language, including the range and use of analogies and metaphors (this is about identification of similarities), and the invention of symbolic language, which includes creating effective notations. We make no claim that these are all the important characteristics; discovering others is the point of our research program.

Key Phrases: *Intelligent Autonomous Systems, Measuring Intelligent Behavior, Constructed Complex Systems, Reflective Infrastructure*

1. Introduction

This paper will describe some characteristics of intelligent computing systems, describe how to make measurements of those characteristics, and discuss what they might mean, though we know that they do not cover the full spectrum of what is commonly considered to be intelligent behavior. We extract these mesaurements from several different viewpoints about what is important for intelligent behavior, and explain their most popularly expected implications.

Intelligence is difficult to measure, because it is thought to be an intrinsic property of systems, like a potential capability or competence, whereas the only things that can be measured are actual performances under various kinds of conditions. This problem has plagued the evaluators of human intelligence since the beginning, to the point that they have generally concentrated on measuring some postulated corresponding performance characteristics [8].

Therefore, metrics can only be based on observed system behavior (though the observations can, of course, measure internal processes from an internal perspective, since we can have some kinds of internal access), since we have no direct access to how internal organization and structure affect intelligence. Even if we assume that intelligence is entirely intrinsic, we cannot evaluate it separately from its corresponding behavior (even if the behavior is only observable introspectively). Measuring performance to infer competence, even of externally observable behavior, is also very difficult and time-consuming, since we intend to use the measurements over a range of situations in order to evaluate the intelligence of different systems.

Success in a particular task is not by itself the right criterion (even if success were well-defined). Many intelligent decisions founder on the rocks of poor information and / or unexpected events, and brute force can make up for a lack of intelligence (e.g., Deep Blue's defeat of Kasparov relied on very fast special-purpose hardware).

Computer programs that play combinatorial games or search the web are not very interesting to us from an intelligent systems point of view, because their domain is so limited and their goals are provided from the outside. Even so, we're interested in computer programs as creative entities (co-investigators, so to speak, instead of just tools), and we think that a careful study of what we can make programs do will be helpful in understanding what the issues are [2] [4]. In order to study these possibilities, we want to define a set of measurements that can be used to differentiate and understand the relationships among different kinds of behavioral characteristics.

We consider autonomy to be more than choosing methods to satisfy goals. A system is autonomous to the extent that it also chooses those goals. In fact, there are really only two classes of (difficult) requirements for effective autonomy: robustness and timeliness. Robustness means graceful degradation in increasingly hostile environments, which to us *implies* a requirement for adaptability, and timeliness means that situations are recognized "well enough" and "soon enough", and that "good enough" actions are taken "soon enough". There is almost never any *optimization* here (that almost always takes too much time and requires too much information).

For the purposes of this paper, we concentrate on the measurement problem instead of the construction problem, though we have some definite ideas about how to build these interesting programs, based on our Wrapping infrastructure for Constructed Complex Systems [17] [21] [22].

2. System Behaviors

We'll start with the assumption that a computing system is designed to help its users _do_ something [9]. That something is a problem in some subject area, such as, for example, copy a file in a computer system, produce a document in a legal office, kill monsters and collect treasures in a computer game, retrieve a web page for a user, solve an equation in a mathematical subject area, find patterns in noisy data in a scientific field, coordinate a distributed simulation for a military application, launch a spacecraft in the aerospace business, collaborate remotely on a design problem for space systems, etc.. We'll use these cases as illustrations in the rest of the discussion.

In all of these cases, there is an *application domain*, which provides a certain context of use and corresponding terminology. Actually, this is more of a *domain-specific language*, since it includes more than just vocabulary terms. It also has a set of abbreviations and conventions about what can remain implicit, and a set of simplifications (which are fruitful lies about the entities and behaviors in the domain). It is important to note that these languages might or might not be written symbolically, since, for example, a computer game is often commanded using a joystick instead of typed commands, and some immersive Virtual Environments are commanded by user movement and gesture.

What the user wants to do is called the *problem*, which only makes sense within the context of interpretation provided by the domain-specific language of the application domain. These languages are used to define the problem context or *problem space*, which is a specialized context within the application domain, in which it makes sense to state a problem.

In other words, it is our opinion that a problem cannot be even stated properly or sensibly without an agreed upon (more often, merely assumed) application domain and problem context. Very often, it is mistakes in the common understanding of this problem context that leads to unexpectedly bizarre or constricting behaviors on the part of the computing system.

So now we have a well-specified problem defined in a problem context. We are purposely setting aside creativity for now, though we believe that this framework can also be applied in that case, with a problem statement of finding the appropriate well-defined problem (this approach is part of our *Problem Posing* paradigm [20]). Explicitly identifying the problem, and separating it from the possible solutions or required user actions, is an important aspect of our approach. It allows many different possible solution methods to be considered. Since NO one analysis or problem-solving method can deal with all problems in a complex domain [6] [1], it is important to have many methods available.

These form the *resource space*, which contains the computational and information resources that are available to address the problem. It is usually implemented as a large set of independent methods, but we think that more structure here can help (which is why we call it a space).

A certain configuration of those resources is needed to address the particular problem that the user has specified. This collection is usually much smaller than the total resource space, so we call it the *solution space*. Since it contains only those resources required to solve the problem, we would ideally like to have the computing system find this space quickly.

However, in order to find a solution space, very often a much larger *examination space* or *discovery space* must be searched.

For example, in trying to prove a theorem (in geometry, say), the problem space is one in which the assertion can be made, the solution space is one in which the proof can be made, and which often involves extra elements constructed just for the proof. The resource space is the collection of lemmas, theorems, inference rules, problem-solving methods, and previously solved problems, and the solution search space is much wider, since it has to include many different kinds of construction and proof discovery methods.

3. Characteristics

In this section, we discuss the following measurable characteristics of intelligent systems (it can be seen that there are nontrivial overlaps among them, which we try to unravel later on):

1. adaptibility,
2. learning,
3. predictive modeling,
4. problem identification,
5. information association,
6. assumptions, and
7. symbolic language.

In each case, we offer an approach to at least one way to compute a measurement value for the characteristic, which we hope will stimulate others to invent and provide better ones.

We make no claim that these are all the important characteristics; discovering others is the point of our research program.

3.1. Adaptibility

By far the most commonly expressed attribute of intelligence is *adaptibility*, which for us means the speed and scope of adaptibility to unforeseen situations, including recognition (of the unforeseen situation), assessment, proposals (for reacting to it), selection (of an activity), and execution. Accurate prediction of effects is even better (and more successful), but we save that one for a later section.

A common example of adaptibility is flexible planning, in which a system can react quickly to situations by changing its plans. It seems clear that flexibility in plans is partly the result of their incompleteness: if the detailed goals remain partly unspecified, then there are more possible steps to take. This phenomenon shows up in programming as "late binding", in which a resource used to address a problem is often not selected until just before it is used (as in our Wrapping approach to heterogeneous system integration in Constructed Complex Systems [19]). The delaying of these decisions does, of course, conflict with rapid execution, and the resulting tradeoff is important and depends essentially on rapid elaboration and evaluations of the choices.

To measure adaptability of a system, we have to present it with different kinds of variability in its environment, and measure its performance, then average that performance over some variability measurement of the environment. The variability in the environment can be static (many different kinds of slowly changing environment), dynamic, (rapidly changing phenomena within the environment), and in both cases, we can describe the degradation in performance as a function of the variability in the environment.

3.2. Learning

Another common attribute of intelligence is *learning*, which for us is the rate of effective learning of observations, behavior patterns, facts, tools, methods, etc. [27]. There is an enormous literature on learning in humans and animals, but our interest here is mainly on the measurements for computing systems that can learn. Learning is about improving performance, so in a sense all of our proposed measurements can be improved by learning. Part of this learning includes concept formation and formulation, which is a way to summarize different structures and processes compactly. We return to this point later on, in the section on symbol systems.

It is important to note here that there are some fundamental limitations on the kinds of symbol systems that can be used in the expressive tasks above. One of the limitations of any discrete symbol system is the "get stuck" theorems [18] [23], which show that unless a system can change its own basic symbols, and re-express its knowledge and behavior in new symbols, new knowledge gradually becomes harder and harder to incorporate, leading to a kind of stagnation.

Measuring learning is a little easier than measuring adaptability. We have long made a distinction between a *smart* system, which has a lot of knowledge about its domain of applicability, and an *intelligent* system, which can learn new knowledge quickly about its domain of applicability. Smartness is a performance characteristic that is relatively easy to measure, and the ability to learn, which is about improving that performance, is easy but time-consuming to measure.

3.3. Predictive Modeling

An important way to be less surprised at environmental phenomena is *predictive modeling*, which for us means accurate modeling and prediction of the relevant external environment. This kind of modeling includes the ability to make more effective abstractions (which is treated below in a later section). Since a system cannot know everything about its environment, we assume that there will be multiple models carried in parallel, with new data interpreted into information using the model as an interpretive context, and each model adjusted, assessed, and ranked for likelihood continually. This kind of modeling makes the computing system an *anticipatory system* in the sense of Rosen [33], since it can make current decisions on the basis of its models of future effects of its decisions. It is therefore expected to be much more capable than a merely reactive system, since it can be preparing responses to its environment before anything important happens in the environment.

A concrete example of this kind of modeling is trying to distinguish trends from fluctuations at different time scales in a complex environment. In such an environment, activity occurs at many time scales, so the only viable approach is multiresolutional [31] [32], that is, the system must maintain several different filtering processes that examine the environment at different resolutions (time, space, and even conceptual), and look for local stationarity.

There are three kinds of models to be considered: empirical models, which are computed according to the observed data, *a priori* models, which are provided up front, and fitted to the data (we think these are much less important than the others), and deduced models, which are derived from other models and knowledge available.

In addition, analyses of these models requires several different kinds of reasoning, both mathematical and linguistic [16]. These methods include *case-based reasoning*, in which the system tries to match the current situation with one it has encountered before, *deductive reasoning*, which can be illustrated as having statements "A" and "A implies B" and concluding statement "B", and *inductive reasoning*, which can be illustrated as having statements "A" and "B" and concluding statement "A implies B". The best-known example of inductive reasoning is *exploratory pattern analysis*, which is a way of extracting properties of mostly unknown data. The last style of reasoning is *abductive reasoning*, which can be illustrated as having statements "B" and "A implies B" and concluding statement "A". This style of reasoning is the one corresponding to explanation, since it follows the deductive chains backwards.

Measuring the modeling capability is not about comparing the resulting models with the processes underlying the environmental phenomena, but rather, it is about measuring the correctness or appropriateness of the predictions. Some predictions take the form "this phenomenon is unimportant", while some must be much more definite, such as "the moving ball will be there at that time" or "the closing door will be open enough for a few seconds". Once explicitly formulated, these predictions can be compared, and the results plotted against the complexity of the prediction task (which we as evaluators must assess).

3.4. Problem Identification

The best way to respond to problems quickly is to identify them quickly, which requires speed and clarity of problem identification and formulation. In our opinion, speed of problem solution is secondary. Even if we seem to specify a problem as a constrained search, we seem to construct search spaces that are very problem-specific, often extremely intricate, constructed using the constraints directly (i.e., not by searching a large encompassing space, and ignoring the parts outside the constraints).

This problem identification problem is a special case of the situation identification problem, in which acceptable performance is often dependent on recognizing that a situation is similar to one encountered before, and that, in turn, depends on identifying the "right" set of features of the situation to explicitly notice and recall.

Naive models of situated computing systems assume that all of the important data that defines a situation is contained in the sensor values for that instant. Humans don't do that; we seem to extract information from the data, based on a number of continual, particular, and only occasionally goal-directed models, and retain only a small part of the actual sensor data. There is also some reason to believe that we only keep interval averages, not instantaneous pictures, of a situation (even a mental image is the result of a lot of processing, for object separation and identification, etc.).

The ability to identify important situation features quickly and correctly depends on having at hand the right specification spaces to determine and describe the features.

Very often, the application domain and problem context that allow a problem or even a situation to make sense must be inferred from the observable environmental behavior. This process is also part of good problem identification, a kind of recognition or noticing.

Good problem identification is an intermediate stage between goals and solutions, so it must in part depend on the resources available to a system.

Criteria for good problem identification are still difficult to describe. We will take speed of problem formulation, succinctness of problem statement, and accuracy of problem statement to be the main criteria. Here, we can only assess the accuracy of the problem statement using knowledge of the potential solution methods, since the effectiveness of the problem statement depends on which resources can address it.

3.5. Association

One of the clearest signs of intelligence is the wide scope and effectiveness of associations, and the corresponding evaluation of disparate information for inclusion into a decision process. Discovery and explanation of new associations is even frequently associated with creativity.

This includes several different kinds of reasoning, from analogies and use of metaphors, through the connection of facts to inference rules. It includes ways to use complex relationships summarized numerically (as we so often do when we implement these systems), and it must include a very flexible reasoning system [16]. There is some argument to the effect that all of these can be viewed as similarities in conceptual spaces [10], as long as we make the class of spaces large enough (i.e., not just numerical ones).

These abstract associations are also part of the mysterious phenomenon of "noticing", which can occur when repeated or anomalous environmental effects are pushed into awareness, seemingly without any prior attention. Similarly, we seem to be adept at noticing correlations in temporal sequences (this ability clearly has some evolutionary advantages), even when they occur in distinct sensory or conceptual spaces.

The simplest version of these processes uses empirical statistical techniques, such as the use of co-occurrence measurements in natural language information retrieval. These and related methods work surprisingly well for this case [26], and we have shown that they can be used in other areas as well [12] [24].

On the other hand, what allows these methods to work well is the explicit representations for words and phrases in the kinds of documents used. In our case of Constructed Complex Systems, the system has to make the representations explicit first, after which the analyses are relatively easy. In particular, it is important to have a representational mechanism that allows comparisons in many different conceptual spaces, so that different kinds of associations can be computed and analyzed.

Since we discuss in other sections the choices of representation and the difficulties of appropriate ones, we consider in this section only the problem of computing associations. We could posit that the wider the associations range, i.e., the more conceptual spaces are involved, the better the association process, but that width of scope has to be traded off against the speed of use of the associations, since we are actually only able to measure performance, not competence. This ability will manifest itself as an improved ability to recognize similarities in difficult problems, and an improved ability to use unlikely resources to address problems in a useful way.

3.6. Assumptions

A perennial problem with reasoning in systems, and particularly with deduction, is the mis-identification and conflation of assumptions. It is important that a system can identify its more important assumptions and prerequisites, which includes the ability to widen a context (by removing some of the assumptions).

This problem is a special case of Computational Reflection [28] [11], which is the ability of a Constructed Complex System to analyze its own behavior [15]. Having access to internal data structures and reasoning processes in an explicit and analyzable way allows a system to monitor its own behavior, short-circuit unsuitable lines of reasoning, and perform "what-if" studies of itself, which can eliminate some errors before they occur [21] [24]. We have shown that it is relatively straightforward to implement systems with this kind of Computational Reflection [17], but the general case is much harder.

We can consider systems that identify the prerequisites of an action, since identification of prerequisites is abductive reasoning (also called "backward chaining" in the Artificial Intelligence literature), but designing a system that can determine a context limitation, which is a kind of prerequisite of representation, and then move outside that limitation, is much harder.

Identifying assumptions is a kind of creative reasoning, that examines reasoned arguments and transform them into an identification of the assumptions and inference rules required to accomplish the arguments. Since we expect the system to perform these operations itself, it must have a mechanism for reasoning within a system, about the boundaries and limitations of that system. We think that this ability is both hard and essential for intelligent systems.

We can measure how well a system identifies its own assumptions by placing it into environments where many common assumptions fail, and checking how well the system performs. We can also use environments in which the basic assumptions change with time, to see if the system can react sufficiently quickly. These measurements are subtle, and disentangling them from the other possible reasons for performance failures will be difficult. We need much better measurements here.

3.7. Symbolic Language

Perhaps the most important property of all, in our opinion, is the use of symbolic language for explicit representations, including the range and use of analogies and metaphors (this is about identification of similarities), and the invention of symbolic language, which includes creating effective notations for internal representation. This property is not altogether unchallenged, but despite the "behavior-based" intelligence work [29], we believe representation to be essential at all levels of intelligence [3], especially for computing systems.

We repeat here that we don't care particularly whether living systems (and in particular humans) have all of these models explicitly represented or implicitly embodied. Our Constructed Complex Systems will have them all explicit.

This property should be unraveled into several different characteristics, but there doesn't seem to be an appropriate analysis of it yet, though there are some promising or at least interesting approaches [34] [7], and we have proposed an architecture that emphasizes the symbol systems [22].

Such an approach to the use of symbols in Constructed Complex Systems must account for the semantics of representation [35], at many different levels, and for the processes that change those representation methods (our conceptual categories are an example representation style [13] [14], and our computational semiotics research is about changing the symbol system when it becomes necessary [18] [23]).

It turns out that human expertise often correlates with better-organized knowledge, and not just with more knowledge, so that problems are recognized more quickly [8].

Since, in our opinion, appropriate abstraction requires a repertoire of conceptual spaces, so that the important properties of the situation at hand can be matched to many more choices of analysis space, and evaluative assessments can become part of the matching process, we think that a very large repertoire is needed, together with some very flexible and fast indexing methods.

Following our own symbol system studies here [13] [14], we measure the use of symbol systems via an efficiency notion: the total size of the representations used compared to the scope of what is represented. This comparison can be estimated using the analysis described in the papers cited: a fixed symbol system has a fixed finite set of basic symbols, and a fixed finite set of symbol structure combination methods. These sets strictly limit the number of distinctions that can be represented within the symbol system with each size expression. If the system can also change the combination methods, then the numbers can be much larger (though they are still computable).

This measurement is, of course, an intrinsic one (i.e., it is a competence measure), not an extrinsic one (i.e., a performance measure), but we think that it will help us develop more performance measurements. In addition, we want some other performance measurements, such as the speed of representational encoding, measured in some units independent of machine-hardware, and the speed of interpretation of those representations (which is about determining the appropriate action to take). There are many other possible measures here.

4. Intelligent Systems

In this section, we discuss how these issues affect the design of Constructed Complex Systems [15], which are artificially constructed systems that are managed or mediated by computing systems. We are concerned with issues of autonomous and intelligent behavior in such systems, which for us, at least means that the system takes a major role in selecting its own goals [17] [25]. When we expect Constructed Complex Systems to operate autonomously, whether out in the real world or in cyberspace, we need to incorporate a great deal of flexibility and adaptability into their design and implementation. We have shown one way to implement such a system [21] [22], one that also helps avoid the most common difficulties found in complex computing systems: rigidity and brittleness.

Biological systems have much more flexible and powerful adaptation properties than most constructed systems [5], and a careful consideration of their properties provides stringent requirements for the kind of Constructed Complex Systems that would be able to act autonomously. It also gives us some hints about the design structures that are needed [30] [17].

Our approach is to define a new kind of architecture [22] that includes both our Wrapping integration infrastructure [19] and our Problem Posing interpretation [20], that provides a declarative interpretation of all programming languages, so that posed problems can be separated from applicable resources, and our conceptual categories [13] [14] to provide a flexible representation mechanism that separates model structures from the roles they play.

Our Wrapping architecture provides the required flexibility by supporting systems that are variable as far down as we choose to make them (even all the way down through the operating system to the hardware) [15]. One reason that we want this variability is that we expect to study many different approaches to any given problem area, and our infrastructure has to support alternatives for almost every part of every process. In fact, one of the principles we have highlighted in our architecture investigations is that NO one model, language, or method suffices for a complex system (or environment), so the variability is not just convenient; it is necessary [6] [1].

In addition, we take the hypothesized common origin of language and movement [3] as a hint, since the implied layers of symbol systems can be implemented easily in Constructed Complex Systems using a meta-level architecture [17].

In addition to the data and processes, we also need a third style of computation, that of "re-expression", which allows a system to re-organize itself when its current organization is not adequate. What this means for us is that the system can somehow detect when its own representational mechanisms are not adequate, and it can use the failures to help invent new ones.

To make things even more interesting, we also want to have the system decide for itself when it needs to be reorganized, because its fundamental symbol systems are not expressive or powerful enough, and then carry out for itself the re-organization automatically, by defining new symbol systems and re-expressing itself in the new terms. This behavior is hard to implement usefully, but we have made some progress in identifying the important issues.

The Wrapping processes give the process structure and the Wrappings and conceptual categories give the data structure. The re-expression criteria are implemented as resources that monitor the system. We describe each of these technical issues in turn, and then show how they can be used to help construct the kind of system we want to build.

The essence of computation is interpretation of symbol systems. The only operations that a digital computer can perform are copying and comparison. All arithmetic in digital computers is via limited-precision explicit models of the corresponding integer or real arithmetic. Therefore, we cannot construct computing systems to do complex or otherwise interesting tasks without many explicit models of the kinds of computation, deduction, or analysis required. All of these models must then be exprcssed in terms of the operations that we can implement on these (very) limited computers.

The theorems of Turing, Go:del, and others show that there are fundamental limits on the expressive and computational power of computing systems, but ALL of the theorems assume that the symbol system remains fixed (that is a basic assumption in all of the mathematical proofs), and that the parallelism can be mapped into interleaved events. Systems that are not restricted in either of these ways might escape the bounds of these theorems. This is one of our current direction of research [18] [22] [23].

5. Conclusions

We care about measuring intelligence because we want to build such devices, and without some better measurement processes, we will have no repeatable way to evaluate and compare different designs.

We have described some properties that we think are important, that have driven our research in Constructed Complex Systems, including a few that have not been extensively used or identified in the literature. We do not think that they completely cover the spectrum of what is commonly considered to be intelligent behavior, but they do cover more of the scope than simply "adaptability" or "intellect".

We have examined these properties to determine what they require as fundamental enabling capabilities, and described an architecture that includes all of these enablers, as a way to test our assertions about the connection between them and intelligent behavior. We expect that as we build systems with more of these enablers, the systems will exhibit more of the important properties we have identified, and at the same time they will seem more intelligent.

We think that this problem is hard, and that we are on a right track (we make no assumption about how many right tracks there may be; the more we collectively explore, the more likely it is that we will get some of the right answers). We think that fundamental investigations like these are necessary; we hope that they are sufficient.

References

[1] Kirstie L. Bellman, "An Approach to Integrating and Creating Flexible Software Environments Supporting the Design of Complex Systems", pp. 1101-1105 in *Proceedings of WSC '91: The 1991 Winter Simulation Conference*, 8-11 December 1991, Phoenix, Arizona (1991); revised version in Kirstie L. Bellman, Christopher Landauer, "Flexible Software Environments Supporting the Design of Complex Systems", *Proceedings of the Artificial Intelligence in Logistics Meeting*, 8-10 March 1993, Williamsburg, Virginia, American Defense Preparedness Association (1993)

[2] Kirstie L. Bellman, "Sharing Work, Experience, Interpretation, and maybe even Meanings Between Natural and Artificial Agents" (invited paper), pp. 4127-4132 (Vol. 5) in *Proceedings of SMC'97: the 1997 IEEE International Conference on Systems, Man, and Cybernetics*, 12-15 October 1997, Orlando, Florida (1997)

[3] Kirstie L. Bellman and Lou Goldberg, "Common Origin of Linguistic and Movement Abilities", *American Journal of Physiology*, Volume 246, pp. R915-R921 (1984)

[4] Kirstie L. Bellman, Christopher Landauer, "A Note on Improving the Capabilities of Software Agents" (poster summary of [17]), pp. 512-513 in *Proceedings of AA'97: The First International Conference on Autonomous Agents*, 5-8 February 1997, Marina Del Rey, California (1997)

[5] Kirstie L. Bellman and Donald O. Walter, "Biological Processing", *American Journal of Physiology*, Volume 246, pp. R860-R867 (1984)

[6] Richard Bellman, P. Brock, "On the concepts of a problem and problem-solving", *American Mathematical Monthly*, Volume 67, pp. 119-134 (1960)

[7] Terrence W. Deacon, *Symbolic Species: The Co-Evolution of Language and the Brain*, Norton (1997)

[8] K. Anders Ericsson, Reid Hastie, "Contemporary Approaches to the Study of Thinking and Problem Solving", Chapter 2, pp. 37-79 in [36]

[9] Kenneth D. Forbus, Johann de Kleer, *Building Problem Solvers*, A Bradford Book, MIT Press (1993)

[10] Peter Ga:rdenfors, *Conceptual Spaces: The Geometry of Thought*, MIT (2000)

[11] Gregor Kiczales, Jim des Rivieres, Daniel G. Bobrow, *The Art of the Meta-Object Protocol*, MIT Press (1991)

[12] Christopher Landauer, "Correctness Principles for Rule-Based Expert Systems", pp. 291-316 in Chris Culbert (ed.), *Special Issue: Verification and Validation of Knowledge Based Systems, Expert Systems With Applications Journal*, Volume 1, Number 3 (1990)

[13] Christopher Landauer, "Conceptual Categories as Knowledge Structures", pp. 44-49 in A. M. Meystel (ed.), *Proceedings of ISAS'97: The 1997 International Conference on Intelligent Systems and Semiotics: A Learning Perspective*, 22-25 September 1997, NIST, Gaithersburg, Maryland (1997)

[14] Christopher Landauer, "Conceptual Categories in the Information Infrastructure", in *Proceedings of ICC'98: 1998 International Congress on Cybernetics*, 24-27 August 1998, Namur, Belgium (1998)

[15] Christopher Landauer, Kirstie L. Bellman, "Constructed Complex Systems: Issues, Architectures and Wrappings", pp. 233-238 in *Proceedings of EMCSR'96: Thirteenth European Meeting on Cybernetics and Systems Research, Symposium on Complex Systems Analysis and Design*, 9-12 April 1996, Vienna, Austria (April 1996)

[16] Christopher Landauer, Kirstie L. Bellman, "Mathematics and Linguistics", pp. 153-158 in Alex Meystel, Jim Albus, R. Quintero (eds.), *Intelligent Systems: A Semiotic Perspective, Proceedings of the 1996 International Multidisciplinary Conference, Volume I: Theoretical Semiotics, Workshop on New Mathematical Foundations for*

Computer Science, 20-23 October 1996, NIST, Gaithersburg, Maryland (1996)

[17] Christopher Landauer, Kirstie L. Bellman, "Computational Embodiment: Constructing Autonomous Software Systems", pp. 42-54 in Judith A. Lombardi (ed.), *Continuing the Conversation: Dialogues in Cybernetics, Volume I, Proceedings of the 1997 ASC Conference*, American Society for Cybernetics, 8-12 March 1997, U. Illinois (1997); poster summary in [4]; *Cybernetics and Systems: An International Journal*, Volume 30, Number 2, pp. 131-168 (1999)

[18] Christopher Landauer, Kirstie L. Bellman, "Situation Assessment via Computational Semiotics", pp. 712-717 in *Proceedings of ISAS'98: the 1998 International MultiDisciplinary Conference on Intelligent Systems and Semiotics*, 14-17 September 1998, NIST, Gaithersburg, Maryland (1998)

[19] Christopher Landauer, Kirstie L. Bellman, "Generic Programming, Partial Evaluation, and a New Programming Paradigm", Paper etspi02 in *32nd Hawaii Conference on System Sciences, Track III: Emerging Technologies, Software Process Improvement Mini-Track*, 5-8 January 1999, Maui, Hawaii (1999); revised and extended version in Christopher Landauer, Kirstie L. Bellman, "Generic Programming, Partial Evaluation, and a New Programming Paradigm", Chapter 8, pp. 108-154 in Gene McGuire (ed.), *Software Process Improvement*, Idea Group Publishing (1999)

[20] Christopher Landauer, Kirstie L. Bellman, "Problem Posing Interpretation of Programming Languages", Paper etecc07 in *Proceedings of HICSS'99: The 32nd Hawaii Conference on System Sciences, Track III: Emerging Technologies, Engineering Complex Computing Systems Mini-Track*, 5-8 January 1999, Maui, Hawaii (1999)

[21] Christopher Landauer, Kirstie L. Bellman, "Computational Embodiment: Agents as Constructed Complex Systems", Chapter 11, pp. 301-322 in Kerstin Dautenhahn (ed.), *Human Cognition and Social Agent Technology*, Benjamins (2000)

[22] Christopher Landauer, Kirstie L. Bellman, "Architectures for Embodied Intelligence", pp. 215-220 in *Proceedings of ANNIE'99: 1999 Artificial Neural Nets and Industrial Engineering, Special Track on Bizarre Systems*, 7-10 November 1999, St. Louis, Mo. (1999)

[23] Christopher Landauer, Kirstie L. Bellman, "Symbol Systems in Constructed Complex Systems", pp. 191-197 in *Proceedings of ISIC/ISAS'99: International Symposium on Intelligent Control*, 15-17 September 1999, Cambridge, Massachusetts (1999)

[24] Christopher Landauer, Kirstie L. Bellman, "Detecting Anomalies in Constructed Complex Systems", in *Proceedings HICSS'2000: The 33rd Hawaii International Conference on System Sciences, Track IV: Emerging Technologies*, 4-7 January 2000, Maui, Hawaii (2000)

[25] Christopher Landauer, Kirstie L. Bellman, "Reflective Infrastructure for Autonomous Systems", in *Proceedings of EMCSR'2000: The 15th European Meeting on Cybernetics and Systems Research, Symposium on Autonomy Control: Lessons from the Emotional*, 25-28 April 2000, Vienna (April 2000)

[26] Christopher Landauer, Clinton Mah, "Message Extraction Through Estimation of Relevance", Chapter 8, in R. N. Oddy, S. E. Robertson, C. J. van Rijsbergen, P. Williams (eds.) *Information Retrieval Research, Proceedings of the Joint ACM and BCS Symposium on Research and Development in Information Retrieval*, June, 1980, Cambridge University, Butterworths, London (1981)

[27] Pat Langley, *Elements of Machine Learning*, Morgan-Kaufmann (1996)

[28] Pattie Maes, D. Nardi (eds.), *Meta-Level Architectures and Reflection, Proceedings of the Workshop on Meta-Level Architectures and Reflection*, 27-30 October 1986, Alghero, Italy, North-Holland (1988)

[29] Maja J. Mataric, "Behavior-Based Control: Main Properties and Implications", pp. 46-54 in *Proceedings IEEE International Conference on Robotics and Automation, Workshop on Architectures for Intelligent Control Systems*, May 1992, Nice, France (1992)

[30] Maja J. Mataric, "Studying the Role of Embodiment in Cognition", pp. 457-470 in *Cybernetics and Systems*, special issue on *Epistemological Aspects of Embodied Artificial Intelligence*, Volume 28, Number 6 (July 1997)

[31] Alex Meystel, "Multiresolutional Architectures for Autonomous Systems with Incomplete and Inadequate Knowledge Representations", Chapter 7, pp.159-223 in S. G. Tzafestas, H. B. Verbruggen (eds.), *Artificial Intelligence in Industrial Decision Making, Control and Automation*, Kluwer (1995)

[32] Alex Meystel, *Semiotic Modeling and Situation Analysis: An Introduction*, AdRem, Inc. (1995)

[33] Robert Rosen, "Anticipatory Systems in Retrospect and Prospect", *General Systems Yearbook*, Volume 24, p. 11 (1979); reprinted as Paper 28, pp. 537-557 in George J. Klir, *Facets of System Science*, Plenum (1991)

[34] Brian H. Ross and Thomas L. Spalding, "Concepts and Categories", Chapter 4, pp. 119-148 in [36]

[35] John Sowa, *Knowledge Representation*, Brooks / Cole, Pacific Grove, CA (2000)

[36] Robert J. Sternberg (ed.), *Thinking and Problem Solving*, Academic Press (1994)

Generalizing Natural Language Representations for Measuring the Intelligence of Systems

A. Meystel
Drexel University, Philadelphia, PA 19104

Abstract. In the core of this method of intelligence evaluation, there is a concept of using natural language as the least damaging medium for representing knowledge of the systems. The goal of all existing methodologies of knowledge representation boils down to performing generalization of this knowledge in one of the existing forms: analytical representation, automata theory, predicate calculus of the first order. Connectionist schemes are not on this list because the problem of generalization upon the entity-relational network (ERN) have not been addressed consistently. In this paper, the concept of constructing a nested multiresolutional system of ERNs by consecutive generalization of them bottom-up and consecutive instantiation of them top-down. It is demonstrated that given a set of problems to be resolved, one can learn which one the nested ERN alternatives is more appropriate for solving this set. Finally, a problem of evaluating ERN "for any set of problems" is discussed.

Conceptual Paradigm. This theoretical paradigm is related to intelligent systems for text processing. Although, it has a broad practical application by itself, we would be interested in considering it as a symbolism for any intelligent system. The goal is to obtain a structure of text organization, elements of which can be used upon the initial narrative for the subsequent processing in order to generate a variety of different texts that have various degrees of compression and/or enhancement. It is anticipated that by constructing a proper organization of the text representation, obtained from the original document, different structures of the text could be constructed, for example, the one that would allow to encode its meaning as a set of nested and interrelated generalizations. In turn, this should allow for generating the narrative text from each of these structures. These texts should be different in their level of generalization, focus of attention, and the depth of detail. Theoretical premises of this paper have been applied for commercial products[1].

General Vision. As soon as the automated analysis starts, the whole texts changes its initial shape and demonstrates a multiplicity of potential interpretations at each level of resolution. The text subjected to the process of multiresolutional analysis demonstrates its semantic fuzziness, and zones of combinatorial possibilities emerge around each unit of the texts[2]. These zones characterize the interpretational ambiguity which should be eliminated (or at least, substantially reduced) as a result of text processing. The fuzzy and not totally disambiguated units have frequently an emergent property of sticking together, forming new generalized units that precipitate from the fuzzy intermediate structure. Eventually a new text emerges which is shorter or longer than the initial one depending on the algorithm applied. The merger of the text units happens in a strictly multi-granular fashion. Each text has a potential to several rounds of compression by generalization as well as to several rounds of enhancement by instantiation. Construction

[1] Cognisphere, Inc., URL http://www.cognisphere.com

[2] The metaphor "combinatorial cloud" alludes to the semantic fields of potential meanings that will depend on our willingness and preparedness to combine various groups of words as potentially salient units of the text.

of such a multigranular structure can be performed for each text, or a group of texts.

The fuzzy zones for combinatorial exploration can be obtained from the text, or can be assigned if the need arises. The process of combinatorial fuzziness generation includes the formation of the links of nestedness, and precipitation of the multigranular text structure (together with levels of enhancement and/or compression). interpretation of an unknown document. However, if the assignment contains the description of a specific customer's interests, this combinatorial fuzziness generation can be guided by this assignment. It does not necessarily need to be guided. In the latter case, a summary of the general (non-goal-oriented) form is created.

Domain of Application. This method of analysis and the computational algorithms are used to obtain a structure of text organization, elements of which can generate a variety of different texts that have different degree of compression.

Each text has a potential to several rounds of compression by generalization as well as to several rounds of enhancement by instantiation. Construction of such a multigranular structure can be performed for each text, or a group of texts (see Figure 1a). The hierarchy of compression by generality can be considered (see Figure 1b) as a set of equally available outcomes.

The expectation is that by moving the representation from narrative to the multiresolutional system of knowledge representation several goals can be achieved simultaneously:

a) evaluation of complexity by determining parameters of the architecture of knowledge

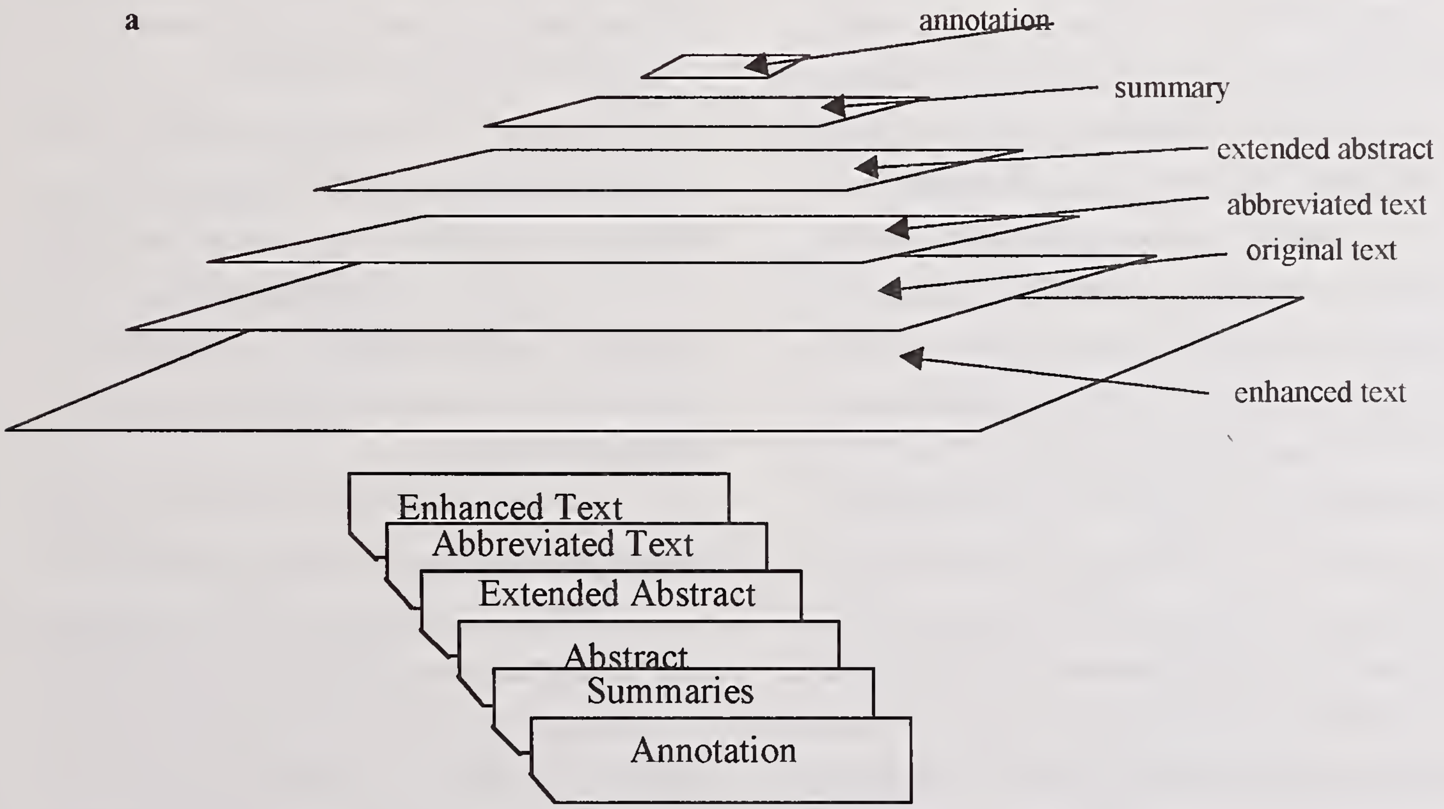


Figure 1. Text and its versions (compressed and enhanced)

representation (number of levels, branching, size of the fuzzy vicinity of the node, etc.).

b) evaluation of the degree of generalization,

c) evaluation of the depth of instantiation,

d) characteristics of the algorithms of focusing attention, clustering and combinatorial search that has generated the architecture.

By using a set of standard text processing routines we succeed in unifying the formal structures of various problems and contexts.

A limited number of standard routines of text processing are used in the proposed method: frequency analysis evaluation of the association strength of the text units and their associations construction of the tentative groups and syntactic parsing are applied as a part of the software package,. These tools play a supportive role in the process of constructing the *multiresolutional structure of text*, the user can use any of existing routines.

Description of processing. Extracting the multiresolutional (multigranular, multiscale) structure (nested hierarchical architecture) of text units (entities) from the Text is a prerequisite to transformation from the narrative representation into the relational architecture of knowledge. The main dictionary is used for the initial interpretation of the units of Text, and the new domain dictionaries are formed for the text-narrative, or Original Text (OT) together with its Structure of Text Representation (STR) as a part of the text analysis. The multiresolutional hierarchy of STR consists of the units, which lump together elements of the text, that has emerged due to the "speech-legacy" grammar. Since the transformation of OT into STR can be done through incremental generalizations within OT, building the vocabulary of the OT is a prerequisite for the subsequent STR construction.

The vocabulary is a list of "speech-legacy" words and multi-word expressions that are symbols for encoding entities of the real situations and can be represented by single words as well as groups of words. An entity of the reality is anything that exists, important for registering and memorizing, has a meaning as a part of some functional description and is (or should be) assigned a separate word (or a group of words) no matter whether we use it as a part of "speech-legacy" representation, or an element of the STR. The first problem to be resolved is finding entities that are represented by single words, then test groups of interrelated words, as phrases that denote entities. Therefore, functioning of STR requires understanding how entities are discovered within Reality (the World): similarly, they will be discovered within the text.

Entities are extracted from Reality as a result of consecutive two-stage testing of the available information (these Stages of processing are universal and are used in all disciplines):

Stage R1. Browsing (searching) the perceived and stored signals and testing the ability to justifiably group them. Creation of hypotheses about groups of the perceived signals (data) that can be interpreted as unified messages that could be put in cause-effect correspondence with other hypotheses about possible groups.

Stage R2. Labeling this hypotheses so that it could be stored and manipulated as a unified group (as an entity) in functioning of the information processing system. Collecting cases of confirmation or rejection for these hypotheses so that for a multiplicity of cases a final decision could be made whether this hypothesis should be confirmed, or rejected.

These stages are performed at any particular granularity, and if there exist more than one granularity, these stages are performed for each of these granularities. If only one level of granularity existed initially, the adjacent level above emerges. The Reality is presumed to contain:

a) entities that we have already learned as a result of prior experiences, and

b) the rest of the Reality that we have not yet learned (whose entities, therefore, cannot be listed); the rest is considered provisionally to be a "uniform" background.

It would be prudent to regard anything unlearned as a continuum, and the process of learning as a process of discovering entities within this continuum. (Continuum is defined as a thing whose parts can't be separated or separately discerned. Initially, the entities hidden in the continuum are indistinguishable). The described approach fits within the scheme of a "scientific method." From various sciences we know that physical laws work in such a way that singular entities are formed from the initially uniform media, and thus, separated from the continuum. These entities are assigned symbols (labels), and they become words in vocabularies.

For each text, the results of structuring can be organized in a hierarchy. Knowing in advance the expected results, let us form and apply the 2-stage procedure.

Text processing oriented toward multigranular structuring is organized in the similar two- stage fashion. We will describe it in more detail. The stages are performed at a particular granularity, and if for the particular text more than one granularity is registered, the stages are performed at each of them. Given a Text, the following operations are performed:

Stage T1. Browsing the text so that all single units of this particular level of resolution could be tested for a variety of the group-forming phenomena. Among the group forming phenomena, the following are of a practical interest as examples of grouping:

-- natural division that provides easily detectable tokens of structuring starting with "Chapters" and ending with "Sentences."

-- frequent spatial adjacency in the text testifies for a possible carrying a particular meaning together; if two words are adjacent, their adjacency might be meaningful, and it would be interesting to see whether these particular two words can be found in this text together again and again; a similar interest might appear about groups of three, four and more words; if a large group of words repeats as a combination many times, it can be considered a possible carrier of meaning "object" or "subject" relevant groups of words should be spatially distinguished from the "action" relevant groups of words; if "nouns" and "adjectives" are swarming together linked with particular rules of grammatical parsing, they can be unified into an ACTOR, or OBJECT-OF-ACTION related groups; on the contrary, if "verbs" and "adverbs" fit within grammatical rules of teaming for Action description – the action-related groups could be detected.

Stage T1*. Creation of hypotheses about groups of the perceived signals (data) that can be interpreted as unified messages that could be put in cause-effect correspondence with other hypotheses about possible groups. We consider them to be a hypotheses about meaningful unit at a particular level of resolution. There are many schools of thoughts that suggest different rules of forming groups because of various grammatical rules and features. They are never 100%

reliable. We will give several examples of using particular rules. However, they should not be considered as a dogma, and for a particular application of the algorithm of structuring multiple additions, subtractions and modifications of rules of grouping should be considered and tested.

"Text" is at the input of the processing. WINDOWING and NEIGHBORHOOD ANALYSIS are performed. First allows for finding all meaningful (frequent) N-word combinations part of which are called here "M-seed." The second serve to analyze with the help of syntactic rules such grammatical couples as "adjective-nouns/pronouns," "nouns-(preposition)-subordinate nouns" and helps to disambiguate difficult cases. As a result, we receive a relational structurc. Somc of the areas in this structure still remain unclear, however (it will be demonstrated later) similar processing at adjacent levels allows to reduce the amount of unclear labels on words and their relationships.

The proccss of "groups hypothesizing" is relevant to each lcvel of granularity. Given explanation above, this diagram is interpreted here for the level of "Words." As a result of groups formation, a new level of generalization is received where the units of thc level are "pair of words" and "triplets of words" and/or "M-seeds" (or the "Seeds of Meaning" that differ by the value of their "significance"). The next level of generalization is combined by further lumping the groups into formations likc: chunks that have a meaning "Being an Actor," or "Action Dcscription," or "The Object Upon Which thc Action Was Directed", etc. Naturally, "group hypothesizing" should be performed at this level too, and the result of this grouping should be simple sentences that are sets as reflected in the actual set of rules. At the next level, we will have grouping into compound sentences. Then, the levels of paragraphs, Sections, Chapters, etc. are going. Running the algorithm of "group hypothesizing" always creates a level above. In Section 4 of this disclosure, this issue is addressed in more detail.

Stage T2. Collecting cases of confirmation or rejection for a particular hypothesis so that for a multiplicity of cases a final decision could be made whether this hypothesis should be confirmed, or rejected. Labeling of the hypotheses is performed provisionally, however after statistical confirmation, the hypothesis is included into the domain dictionaries permanently. After this it can be legitimately manipulated as an entity and participate in functioning of the information processing system. All obtained groups are tested by using the rule-base that detects N-word groups having relationships of the type: "adjective with its noun," main noun-preposition-subordinate noun", and others. This refinement plays a decisive role in finalizing the text structure.

Units of representation generated within texts. The decomposition of the uniform chaotic informational medium takes place driven by the initial goal and a set of criteria that might determine different kinds of uniform media. Thus, the results of developing the linguistic world representation depends on the aspect of interest submitted and encoded by the user. As a result of recognition processes, a variety of singular information units (entities) emerges, which fit within a natural categorization that is implicitly influenced by the observer. Formation of singularities (as entities) can be metaphorically described as a result of clustering processes in which the elementary units of the primordial Text gravitate to each other in the areas of higher informational density (where the elementary units are more in quantity, more interrelated, and more important for the user. For clarifying the gravitational metaphor, we should emphasize that for the further discussion it is irrelevant whether the density is increased as a result of the gravitation, or gravitation starts prevailing because of an initial increase in density. In our disclosure, these processes are to be understood in computational terms. At this point, the observer will legitimately appear in our presentation as a carrier of the interrelated concepts: scale, resolution, and granulation. ("Resolution" is determined in the same way as the "granule" by the size of the smallest distinguishable zone, a pixel, or a voxel[3] or even a "word" of the space in which we describe our system. Scale is a value inverse to the resolution.)

The concept of scale allows for introduction of a formidable research tool that can be applied for each couple of adjacent levels of knowledge organization obtained by the method described in this Section of the disclosure. This tool is related to the specifics of a different interpretation of units in higher and lower levels of resolution (HLR and LLR). The units of the HLR emerge as a result of the process of forming singularities at the previous, even higher resolution level (which is not a part of our couple levels of resolution under consideration). After these singularities have been formed, they receive an interpretation, a meaning, a separate word of a vocabulary at this HLR. For the LLR of the pair of levels that we discuss, these particular singularities have no meaning at all—not yet!

The meaning will emerge after these entities of the higher level of resolution (HLR) will assemble together into a singularity which can be recognized by the user at the lower resolution level (LLR) as a meaningful entity. Before grouping of these entities into meaningful singularities happens, they are just nameless units with a tendency to gravitate to each other, expressed in the set of their relations. This phenomenon is similar to physical gravitation although the gravitation "force" depends on the text, context, goals, and other details of the situation. So, the process of entity formation for LLR recognizes the entities of a HLR just as a set of anonymous units. Their "gravitational" field leads to clustering of features and can give a birth to a new entity of LLR.

Uniform (chaotic) medium is always a collection of some non-uniform units at a finer scale (at higher resolution), and uniformity of the medium is a parameter that we obtain from characterizing the medium at a coarser scale (at lower resolution). In order to compute this parameter (the degree of uniformity, or density) different techniques can be used. All of them work as follows.

Phenomena of Attention: Scope and Focus. Windowing is a result of the need to focus our attention within a specific scope. Let us consider a particular zone of the medium that we use to evaluate; we will call it *the scope of interest*. An imaginary large window (*the scope of attention*) is to be imposed upon the medium (*scope of interest*). Then, the smaller window is sliding within the scope of interest to evaluate the information density. Thus, the size of the scope of attention is presumed to be substantially smaller than the scope of interests. Density of non-uniform units is to be computed within this window which allows evaluation of the continuum quantitatively. Then the window slides

[3] "Pixel" is the smallest indistinguishable unit of a two-dimensional surface. "Voxel" is the smallest spatial indistinguishable unit of a three-dimensional volume. Frequently, the term "voxel" is used in the N-dimensional case. In the single-dimensional case we are talking about a unit of the scale.

over the whole scope of interest, and in each position the density is again computed.

The sliding strategy of moving the window of attention over an Image and/or Text is assigned in such a way that all scope of interest can or will be investigated efficiently. This strategy can be different for constructing different models: we can scan it in a parallel manner; we can provide a very unusual law of scanning; we can make random sampling from different zones of the scope of interest. The strategy selection should depend on needs, hardware tools, and resources available (for example, time). If values of density are about the same everywhere (with small variations within some particular interval) then the medium is considered to be uniform.

Notice, that

a) that in order to introduce the concept of uniformity we used a sliding window which is one of the techniques of *focusing attention;*

b) that in order to form entities of a particular level of resolution we should *group* the entities of the higher level of resolution;

c) that to find candidate units for grouping we should *search* for future members of these groups or otherwise *combine* them together. Later we will return to these operations as components of the elementary unit of intelligence.

More details on the nature of our approach can be found in [1-6].

References

1. A. Meystel, *Semiotic Modeling and Situation Analysis: An Introduction*, AdRem, 1995
2. A. Meystel,"Multiresolutional Semiotic Systems", Proc. 1999 IEEE Int'l Symposium on Intelligent Control, Boston, MA 1999, pp. 198-202
3. A. Meystel, "Multiresolutional Text Interpretation System", In the collection, *Sign Processes in Complex Systems*, publ. by the 7th International Congress of IASS/AIS, Dresden, Germany, October 6-11, 1999
4. A. Meystel, "Multiresolutional Umwelt: Toward Semiotics of Neurocontrol", *Semiotica* , No. 3-4, 1998, pp. 343-380
5. "Semiotics: The Toolbox of Intelligence", in Proceedings of the 1997 International Conference on Intelligent Systems and Semiotics, Gaithersburg, MD, 1997, pp. iii-vi
6. A. Meystel, S. Uzzaman, "Multiresolutional Decision Support System, " US Patent 6102958, 2000

Towards Measures of Intelligence Based on Semiotic Control *

Dr. Cliff Joslyn

Computer Research and Applications Group (CIC-3)
MS B265, Los Alamos National Laboratory
Los Alamos, NM 87545
joslyn@lanl.gov, http://www.c3.lanl.gov/˜joslyn

July, 2000

Abstract

We address the question of how to identify and measure the degree of intelligence in systems. We define the presence of intelligence as equivalent to the presence of a control relation. We contrast the distinct atomic semioic definitions of models and controls, and discuss hierarchical and anticipatory control. We conclude with a suggestion about moving towards quantitative measures of the degree of such control in systems.

1 Introduction: A Control Theory Framework for Intelligence

We consider some of the challenges presented in the white paper designed to prepare for this conference [13]. I take the fundamental question to be "How can we as external observers measure the degree of intelligence in a target system?"

One approach is to invoke the typical lists which can characterize intelligent behavior, including adaptability, complexity of internal models, problem solving ability, etc. But what is fundamental to each of these? For example, adaptability is the ability to adjust responses to make them appropriate under variable conditions. Problem solving is the ability to come to a correct choice about actions to achieve a particular goal, hereby solving the problem. And finally, complexity of internal models must always be considered as relative to their ability to predict the outcome of future behaviors.

Thus can see that fundamental to all of these is the idea that intelligence requires the ability of a system to make *appropriate decisions given the current set of circumstances* [1, 2, 3]. On analyzing this a bit further, we can identify the following necessary components:

Measurement: The ability to know the current set of circumstances.

Decision: The freedom to choose between one of many posibilities.

Goal: The possibility that the choice made will be either appropriate or inappropriate relative to a goal state.

Action: The ability for the decision to affect external and future events, in order for them to be either closer to or further away from the goal.

2 Intelligence as Semiotic Control

We note the similarity to the scheme of an intelligent system as outlined in the conference White

[*] Prepared for the 2000 Workshop on Performance Metrics for Intelligent Systems.

Paper [13]. This requires a "loop of closure" consisting of six modules: a world interface, sensors, perception, a world model, behavior generation, and actuation. We understand this situation as the existence of a *semiotic control* system. We know briefly outline the theory of semiotic systems.

2.1 Semiotic Models and Controls

There is a rich literature (eg. [5, 15, 17, 18, 19]), traceable back to the founders of systems theory and cybernetics in the post-war period [4], which has tried to construct a coherent philosophy of science based on two fundamental concepts:

- **Models** as the basis not only for a consistent epistemology of systems, but also as an explanation of the special properties of living and cognitive systems.
- **Control systems** as the canonical form of organization involving purpose or function.

While controls and models are distinct kinds of organization, what they share is a common basis in semiotic processes, in particular the use of a measurement function to relate states of the world to internal representations. Perhaps for this reason there has been some ambiguity in the literature about the specific nature of controls and models, and more importantly how the interact. This has led to confusion, for example, about the role of feedback vs. feedforward control, and endo-models *within* systems vs. exo-models *of* systems.

Consider first a classical control system as shown in Fig. 1. In the world (the system's environment) the dynamical processes of "reality" proceed outside the knowledge of the system. Rather, all knowledge of the environment by the system is mediated through the measurement (perception) process, which provides a (partial) representation of the environment to the system. Based on this representation, the system then chooses a particular action to take in the world, which has consequences for the change in state of the world and thereby states measured in the future.

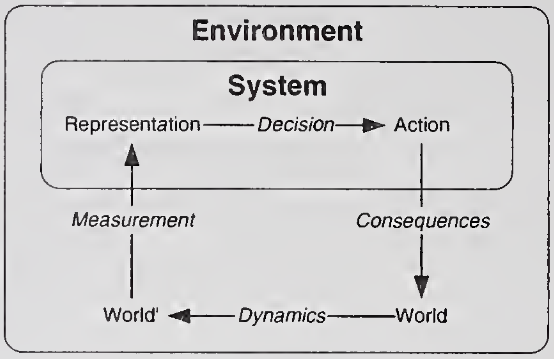


Figure 1: Functional view of a control system.

To be in good control, the overall system must form a negative feedback loop, so that disturbances and other external forces from "reality" (for example noise or the actions of other external control systems) are counteracted by compensating actions so as to make the measured state (the representation) as close as possible to some desired state, or at least stable within some region of its state space. If rather a positive feedback relation holds, then such fluctuations will be amplified, ultimately bringing some critical internal parameters beyond tolerable limits, or otherwise exhausting some critical system resource, and thus leading to the destruction of the system as a viable entity.

Now consider the canonical modeling relation as shown in Fig. 2. As with the control relation, the processes of the world are still represented to the system only in virtue of measurement processes. But now the decision relation is replaced by a prediction relation, whose responsibility is to produce a new representation which is hypothesized to be equivalent (in some sense) to some future observed state of the world. To be a good model, the overall diagram must commute, so that this equivalence is maintained.

As outlined here, models and controls are distinct and atomic kinds of organization. We have argued [8] that this capability begins with living systems, and perhaps defined the necessary and sufficient conditions for living systems.

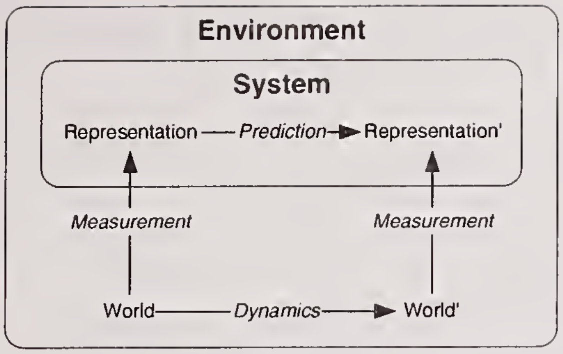


Figure 2: Functional view of the modeling relation.

2.2 Hierarchical Control

Of course, all of the relations described here are a great deal more complex in real intelligent systems. In particular, usually controls and models are considered together. This concept is fully developed elsewhere [7, 9]. We now summarize the primary results of these considerations.

First, the classical view of linear control systems theory [14] is recovered by introduced a "computational" step which plays the role of cognition, information processing, or knowledge development. Typically, extra or external knowledge about the state of the world or the desired state of affairs is brought to bear, and provided to the agent in some processed form, for example as an error condition or distance from optimal state. So now measured states are manipulated and compared to a goal state.

In particular, we are impressed by Bill Powers system for hierarchical control [15, 16, 6], which he has succesfully generalized to explain the architecture of neural organisms. As shown in Fig. 3, he views the computer as a comparator between the measured state and a hypothetical set point or reference level (goal). This then sends the second representation of an error signal to the agent. He also explicitly includes reference to the noise or disturbances always present in the environment, against which the control system is acting to maintain good control. For us, these are bundled into the dynamics of the world.

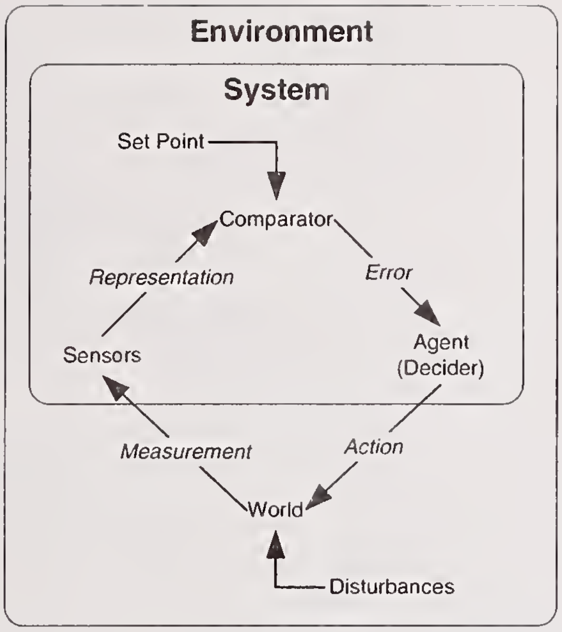


Figure 3: A Powers' control system.

Another great virtue of Powers' control theory model is its hierarchical scalability. Fig. 4 shows such a hierarchical control system, containing an inner level 1 and the outer level 2. The first key move here is to allow representations to be combined to form higher level representations. In the figure S_1 and S_2 are low distinct level sensors providing low level representations R_1 and R_2 to the inner and outer levels respectively. But R_1 is also sent to the higher level S_3, and together they form a new high level representation R_3.

The second step is the ability for the action of one control system to be the determination of the set-point of another, thus allowing goals to decomposed as a hierarchy of sub-goals. In the figure, the outer level uses R_3 to generate the action of fixing the set point of the lower level. Note how this recovers Meystel *et al*'s "Feature 10" of multiscale knowledge representation where the action of a lower level system is actually the goal of an upper level system [13].

Notice also that the overall topology of the control loop is maintained. While ultimately the lower level is responsible for taking action in the world, it is doing so under the control of the comparison of a high-level goals against a high-level representation. Neural organisms especially are

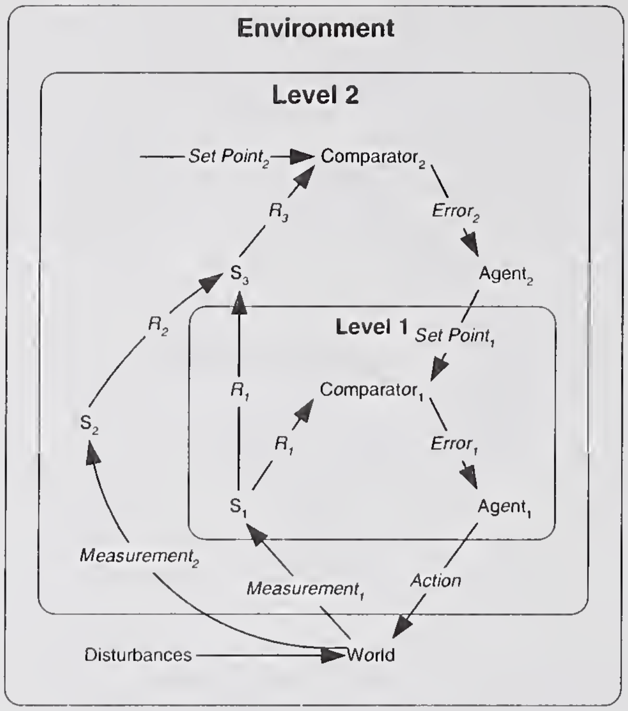


Figure 4: Hierarchical nesting of Powers' control systems.

systems of this type, low-level motor and perceptual systems combining to accomplish very high-level tasks. And of course, determination of the outermost goal is not included within Powers' formal model.

2.3 Anticipatory Control

While familiar to us as a standard engineering discipline, a number of researchers are pursuing the applicability of this kinds of semiotic control [12]. It is also being generalized to a number of other engineering [2] and scientific domains.

However, our normal sense of control combines it with models, which are used to aid in decision-making by predicting future states of anticipated actions, using prediction of future events to guide actions. This is what Ashby refers to as “‘cause control” [4], or Rosen as “anticipatory” [17], or Klir as feedforward [10]. In this architecture an endo-model embedded *within* a control system is used to make a decision as to which action to take, and thus acts in the role of the agent. It is this view which most dominates our conception of the nature of control in general.

However, this architecture is actually highly complex and special. It is shown in Fig. 5, where now the agent is replaced by an inner system which is *both* a model and a control system (the arrows have been reflected diagonally to make the graph planar and ease the drawing). This inner system is a control system in the sense that there are states of its “world”, its “dynamics”, and an “agent” making decisions.

However, it is also a model in that the states of its “world” are in fact representations, and its “dynamics” is actually a prediction function. The inner system is totally contained within the outer system, and runs at a much faster time scale in a kind of modeling “imagination”. The representation R from the sensors is used to instantiate this model, which takes imaginary actions resulting in imaginary stability within the model. Once this stability is achieved, then that action is exported to the real world.

Note that the outer control loop here is simple, lacking computation. In Powers' terms, there is no set point which the state of the internal model is being compared to. But this could be present in a slight elaboration where an imaginary measurement is taken from “world′” and compared to some set point. The outer error signal would then be fed to change the imagined actions inside the model until stability is achieved.

3 Tests for the Presence of Control

Thus we have now transformed the original question of “how do we measure intelligence?” to “How can we as external observers determine whether a target system manifests control relations with its environment?” and “How can we then measure the degree and modalities of that relation?” I would then offer some ideas based on the work of Powers and his colleague Rick Marken [11, 15, 16].

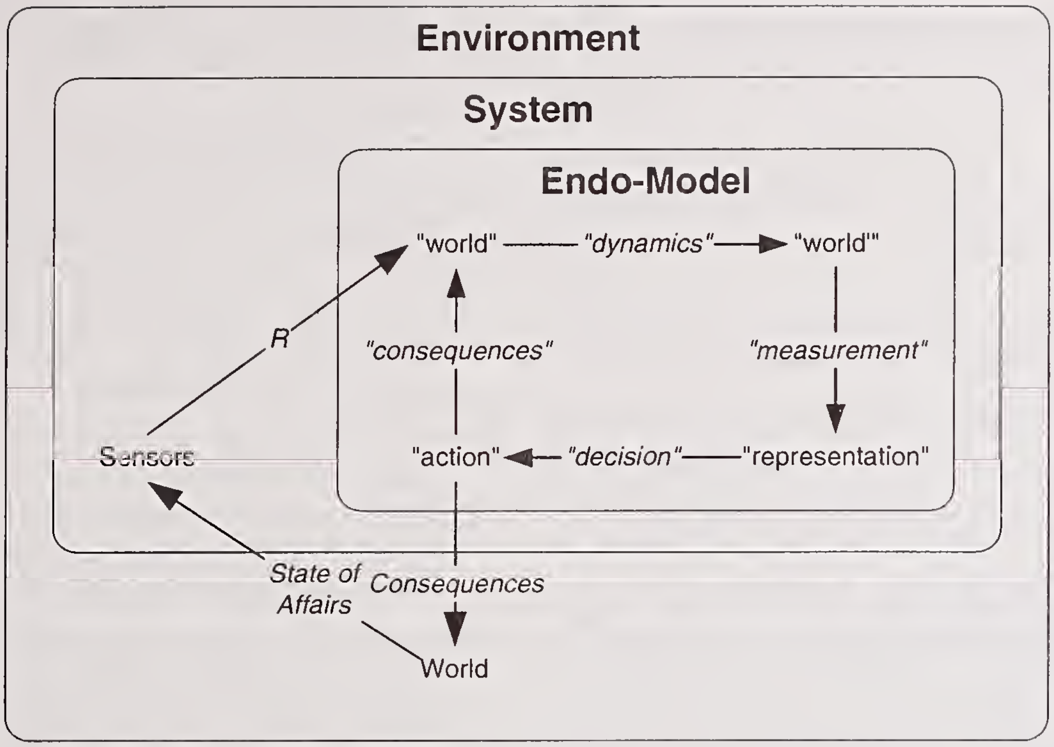


Figure 5: Anticipatory control.

They address the question from the following perspective. Control relations, in virtue of the stability of the controled variables in the environment, have many of the characteristics of other equilibrium phenomena. Both the thermostat and the ball rolling to a stop at the bottom of a hill evidence this kind of stability behavior. In the first case, the ball does not *want* to roll down the hill, but in a very real sense, the thermostat *does* want to regulate its "perception" of the state of the room temperature.

So how can we distinguish a complex dynamic equilibrium from a control relation? Powers and Marken do this distinguishing on the basis of what they call The Test. It involves the system acting in a way which is counter to physical law: if the ball *failed* to roll down the hill, we'd be surprised, thus we hypothesize that such a ball is manifesting a control relation. Similarly, we would normally expect a room to come to equilibrium with its environment. When it does not, and we believe our dynamical model, then we would hypothesize the presence of a control device, and we might investigate and discover a thermostat. The "intelligence" of such systems is based on their manifesting a semiotic relation which has been selected by evolution or by designers, allowing the system to "choose" to act counter to physical law.

Now the rub is that this Test thereby requires the prior presence of a model of what the system *should* be doing, so that we can be surprised when it fails to do so. Thus our recognition of a control relation in an exogenous system requires of us an *exogenous* model of reality, whether or not the system has any *endogenous* model itself.

4 Towards a Measure of Control-Based Intelligence

So now, given this semiotic control-based view of intelligence, we wish to go on and attempt to quantify and characterize the degree and kind of control relations present. Thus the problem of measuring intelligence revolves around our ability to measure:

- The amount of phenomena under control;
- The number of environmental distinctions measured by the system;
- The complexity of modalities of measurement and control;
- The complexity of the environmental variety available to the measurement and control of the system;
- If hierarchical control is present, what is the depth of the hierarchy of control; and
- If anticipatory control is present, what is the complexity of the internal, endogenous models?

No doubt in both real and designed systems these are all related to each other in complex ways. However, each of these quantitative terms is effectively a statistical information measure, a measure of variety or freedom. Thus th are ammenable to information-theoretical measures like entropies, based on quantities of variety, distinctions, and constraints which a control system can recognize in its environment and then act on in appropriate ways.

References

[1] Albus, James S: (1991) "Outline of a Theory of Intelligence", *IEEE Transactions on Systems, Man, and Cybernetics*, v. **21**:3, pp. 473-509.

[2] Albus, James S: (1999) "The Engineering of Mind", *Information Sciences*, v. **117**, pp. 1-18

[3] Albus, James S: (2000), discussion on email list `iab-list@icompsol.com`

[4] Ashby, Ross: (1956) *Introduction to Cybernetics*, Methuen, London, `http://pcp.vub.ac.be/books/IntroCyb.pdf`

[5] Cariani, Peter A: (1989) *On the Design of Devices with Emergent Semantic Functions*, SUNY-Binghamton, Binghamton NY, PhD Dissertation

[6] *Control Systems Group*, `http://www.ed.uiuc.edu/csg`

[7] Joslyn, Cliff: (1995) "Semantic Control Systems", *World Futures*, v. **45**:1-4, pp. 87-123

[8] Joslyn, Cliff: (1998) "Are Meaning and Life Coextensive?", in: *Evolutionary Systems*, ed. G. van de Vijvier, pp. 413-422, Kluwer

[9] Joslyn, Cliff: (2000) "The Semiotics of Control and Modeling Relations in Complex Systems", under review for *Biosystems*

[10] Klir, George: (1991) *Facets of Systems Science*, Plenum, New York

[11] Marken, Richard S: (1988) "The Nature of Behavior: Control as Fact and Theory", *Behavioral Science*, v. 33, pp. 196-206

[12] Meystel, Alex: (1996) "Intelligent Systems: A Semiotic Perspective", *Int. J. Intelligent Control and Systems*, v. **1**, pp. 31-57

[13] Meystel, Alex, *et al.*: (2000) "Measuring Performance of Systems with Autonomy: Metrics for Intelligence of Constructed Systems", white paper for *2000 Workshop on Performance Metrics for Intelligent Systems*, `http://www.isd.mel.nist.gov/conferences/performance_metrics/white_paper.html`

[14] Nise, Norman S: (1992) *Control Systems Engineering*, Benjamin-Cummings, Redwood City CA

[15] Powers, WT: (1973) *Behavior, the Control of Perception*, Aldine, Chicago

[16] Powers, WT, ed.: (1989) *Living Control Systems*, CSG Press

[17] Rosen, Robert: (1985) *Anticipatory Systems*, Pergamon, Oxford

[18] Rosen, Robert: (1991) *Life Itself*, Columbia U Press, New York

[19] Turchin, Valentin: (1977) *Phenomenon of Science*, Columbia U Press, New York

Selected Comments on Defining and Measuring of Machine Intelligence

Paul K. Davis
RAND and RAND Graduate School
1700 Main Street
Santa Monica, CA 90407-2138
June 11, 2000

ABSTRACT

This paper records some thoughts about defining and measuring machine intelligence. It touches on (1) the shortcomings of any scalar metric; (2) the power of having mixes of intelligence types in a population of machines; (3) the special issues related to "common sense;" (4) the need to broaden discussion beyond normally understood intelligence; (5) consistent with that, the need in a population to assure for exploration and "mutation;" (6) some technical issues in modeling reasoning in agents; and (7) a methodology (exploratory analysis) for measuring intelligence that emphasizes a diversity of contexts.

KEYWORDS: *agents, exploratory analysis, multiresolution modeling, machine intelligence, robot intelligence*

1. INTRODUCTION

This is an informal think piece recording a number of thoughts generated by a lengthy discussion, accomplished by e-mail, in preparation for a conference on defining and measuring machine intelligence. Section 2 argues against the search for any simple metric for intelligence. Section 3 draws upon experiences in other domains to suggest that intelligent machines might be assessed as groups exhibiting a good *mix* of intelligences types. Section 4 asks what we might seek in such groups of machines, notes that narrow concepts of intelligence do not obviously allow for wisdom, and relates the search for wisdom with meta-knowledge subjects such as ethics. Section 5 touches upon the issue of learning and ties this to the need, in some populations of intelligent machines, for attributes encouraging exploration and discovery. Section 6 discusses the need for intelligent machines to have internal models of their external worlds (and perhaps themselves); it then summarizes briefly some potentially relevant lessons learned from my own work with multiresolution, multiperspective modeling (MRMPM) of decision making. Finally, Section 7 suggests a new approach or measuring intelligence, an approach that would use emerging concepts and techniques for "exploratory analysis" to assesses a machine's (or group's) effectiveness over an enormous range of conditions.

2. NO SINGLE METRIC MEASURES INTELLIGENCE

2.1 The Multiple Dimensions of Intelligence

One of the many lessons learned from a century of work on human intelligence is that intelligence is multifaceted [1],[2]. This, of course, accords with our everyday observations as well, but the focus—by scientists and organizations—on a single metric of intelligence (IQ, or related items such as college-board scores), has arguably done a great deal of mischief and interfered with what might otherwise have been natural: recognizing and honoring the richness of human capabilities that we refer to collectively as "intelligence." It appears wise to define and measure machine intelligence as a multidimensional concept from the start.

To say that the *functionality* of intelligence is multidimensional (and multifaceted) does not necessarily mean that the underlying capabilities manifesting themselves as functional intelligence are fundamentally different. Indeed, it has been argued that a single set of abstractly-characterized capabilities (e.g., search) applies across the range, and that intelligent behavior—across domains—can be seen as a common set of nested behaviors [3], [4]. That may well be the case; if so, it is a matter of enormous significance. However, my point here is that in seeking ways to measure the intelligence of machines we will often be looking at functional behaviors (e.g., the accomplishment of tasks) and we should not should not make the mistake of imagining that functional intelligence can usefully be reduced to a single metric.[1]

2.2 Connections with Multi Attribute Utility Theory

Some may quarrel with this conclusion. After all, in many endeavors it has proved feasible to combine various factors into a single scalar quantity that reasonably measures what we are interested in and proves quite useful. We see this in the applications of multi-attribute utility theory (MAUT) [5],[6] and in countless modeling problems where people introduce

[1] Gardner's types are linguistic, logical-mathematical, bodily-kinesthetic, spatial, musical, interpersonal, and intrapersonal.

abstractions that combine various factors in ways that appear adequately sound.

Based on some decades of experience, we now know a great deal about the usefulness and shortcomings of MAUT, although the subject is still one that divides people of good will. Personally, I urge students of policy analysis to *savor* the multiple attributes of strategies and avoid combining them until and unless it is necessary. The paradigm for displaying results of policy analysis is, for most of my colleagues and me, a "scorecard' in which one views the ratings of options in each of a number of aggregate categories. We may or may not add up the scores for the purpose of having a single, simple-minded, result (e.g., for making cost-effectiveness comparisons), but if we do it is only after we have adjusted assumptions so as to assure that the aggregated result is "right." By that I mean that decision-analysis methods are often most useful when used iteratively: we try to be logical and explicit; we try to do things by the numbers; we look at the results; we then observe that they are "wrong" (meaning that we don't like them). We go back to the assumptions and either fiddle the input scores or muse a bit until we discover some hidden variables that are bothering us, and affecting us implicitly. We then iterate. And so on.[2]

At the end of the process, the algorithms may work and we may have a sense that we understand the problem, but this was due to the disaggregated process of getting to that point. At the trivial level, I like to challenge students with a car-buying problem, the purpose of which is to demonstrate that the usual hard-headed approach does a *terrible* job in representing our real values. Some individuals, for example, really do want a red Mercedes sportscar, and it's hard to get that answer when looking at mileage and repair costs. Even if one has a category for prestige or some such, it is very difficult to get the red Mercedes as the answer unless one essentially zeros out the other categories or recognizes the shortcomings of the MAUT methodology with its assumptions of linearity and related substitutability.[3]

The value of not abstracting early to a single number is evident to all of us who use Consumer Reports or other commercial reviews of competing products. The approach has won in the marketplace. It is therefore a bit puzzling to me why the allure of the single metric continues to be so high in some technical domains, and why reductionist versions of decision analysis are so commonly taught (with only lip service to the caveats).

2.3 The Persistent Tyranny of IQ and SAT Scores

Despite the shortcomings of single-metric methods, the fact is that IQ scores, SAT scores, and GRE scores are ubiquitous in assessments ranging from the personal ("Wow, Marcus is really smart: he got a double 800.") to hard-nosed decisions by admissions committees at universities and managers in industry. We all know that intelligence is a complex issue, but most of us nonetheless use the simple metrics—at least to some extent. Moreover, they are more than *pure* crutches or scientific astrology; i.e., they actually do correlate, at least to some extent, with things we care about (e.g., performance in classes or in the business environment). And shorthands are useful.

This said, the quality and depth of any discussion of machine intelligence and its measurement would likely be greatly restricted by having a reductionist goal such as finding a single "IQ." Talking in shorthand is an excellent way to "dumb down" conversations and inquiries.

Consider the following as part of an indictment [see also [2]):

- The correlation of IQ and SAT scores with subsequent performance in graduate school and life is modest. Indeed, it is so modest that one can only puzzle about why so much fuss is made over the related tests. The answer appears to be that, bad as the scores are as predictors, they are the best available. [8]

- The predictive power of the simple metrics is particularly poor in explaining, for example, the effectiveness of top executives.[4]

- We probably all know individuals who would flunk tests of mathematics, but who are brilliant in other ways—whether verbally, or, for example, in the arts.

- Most of us know individuals who scored very highly on intelligence tests and yet lack the capability to excel in

[2] See [7] for an example of scoreboard methods applied in one of my recent projects for the defense department. This methodology includes what amounts to multi-attribute utility theory in providing the opportunity to calculate the rand-ordered goodness of options as measured by a composite cost-effectiveness, but the central focus is a more disaggregated scorecard with multiple measures. Further, the methodology emphasizes exploratory analysis across different assumptions and values. This is in contrast with more usual decision-analysis methods that seem to emphasize getting the inputs "correct."

[3] In principle, MAUT theory (e.g., [5], [6]]) allows for nonlinear combining rules and includes methods for inferring values systematically. However, these add a good deal of complexity and tediousness.

[4] It is perhaps of interest that the SAT scores of both Presidential candidates in 2000 were bandied about in the press. Neither candidates scores appear outstanding when compared to those of top-half applicants to graduate schools. Given the achievements of the individuals to date, doesn't this tell us something?

various higher level activities. Perhaps they lack common sense; perhaps they lack creativity; perhaps they are so obsessed with numbers that they cannot deal with fuzzier aspects of life.

Or, on the other end, we can all think of "geniuses" who behaved in ways that can only be regarded objectively as *stupid*. My favorite example is Napoleon, who marched on Moscow in the winter and predictably lost nearly his entire army.

In summary, we should avoid with prejudice the goal of finding a simple-minded metric for robots such as IQ. Multiattribute utility theory will not suffice either—except in highly controlled circumstances—because reality is much too nonlinear and complex.

3. THE POWER OF MIXES

Once we recognize that intelligence is a multifaceted concept, and that society places a high value on all aspects of intelligence, broadly construed, then we are also ready to recognize the value of healthy *mixes*:

- Instead of optimizing the average "IQ" of a robot community, we should instead seek to "optimize" the effectiveness of the community (perhaps omitting those items we expect or want humans to continue to do).

Moreover, in "optimizing," we should apply nonlinear schemes that assure that we don't end up with able mediocrities. For example, in human society most of us believe that we benefit from having at least some people who are extremely good at mathematics, physics, written verbal matters, spoken language, the arts, and even the difficult human skills associated with the very best of leaders on the one hand, or the best of clinical psychologists on the other. But we don't require all of these skills from everyone.

To use a different analogy, consider how we go about dealing with medical issues. Perhaps some readers have a single physician who "does everything" from delivering babies to extracting brain tumors, but the rest of us seek to have a mixture that includes top diagnosticians (the best of whom are very smart in the traditional sense), very good internists who deal more with quantity than the with the hardest cases, and various and sundry specialists. Some of the specialists may be superb at some skills (e.g., microsurgery), but poor at others. Whether fair or not, the stereotype of surgeons is one of blockheadedness, arrogance, and inability to deal with human issues or even medical subtleties that are not "mechanical." Many surgeons even kid about this, describing themselves as world-class plumbers. Now, suppose that we wanted to choose a mix of doctors for a community on the moon. Would we look for some metric, test everyone, and then optimize, or would we instead identify many attributes and assure that all were adequately represented?

4. THE CHALLENGES OF WISDOM AND COMMON SENSE

Despite familiarity with the hilarious (or infuriating) shortcomings of some artificial intelligence programs, I am not particularly mystical about issues such as wisdom and common sense. Intuitively, I believe that they have to some extent been over-rated as a reaction to failures of the straightforward rule-based approaches in AI. I suspect that with large enough computers and sufficient emphasis on and time spent in training with neural nets and other technologies, machines will eventually have moderately good skills that include what look like wisdom and common sense over significant domain areas. On a sliding scale, I probably place myself closer to Raymond Kurzweil in this regard, than, say Herbert Dreyfus.

Nonetheless, this remains a frontier area for research. Measurements would depend not just on the intelligence "wired in," but the intelligence developed by experience and the data bases provided initially and built up over time. As we know from discussions in many forums, it is notoriously difficult at present to measure the information, knowledge, or value in data bases. This, then, is just a warning of a different type.

As a related matter, we should avoid the error of focusing exclusively on aspects of intelligence distinct from ethics, morality, or spiritualism (broadly construed). It is of interest to note that this mistake was *not* made by the late, great Isaac Asimov. It was not accidental that Asimov, rather than more pedestrian writers, took on these issues directly.

We know that one of the special characteristics of intelligent people is that they learn, taking on knowledge and skills that go beyond what they were "programmed for." However, without some kind of principles to act as filters, what machines (or, for that matter, people) choose to learn and experiment with may prove dangerous. Again, we can look to science fiction for examples

As a hypothesis, it seems to me that

- It will continue to prove impossible to achieve top-notch "intelligent performance" across a wide range of situations without having principles that look more like ethics than electrical engineering.

5. ASSURING EXPLORATION AND MUTATION

Although we may differ among ourselves about the meaning and existence of "progress," it is clear that the processes of evolution such as mutation and natural selection have profound effects. Suppose that there were no mutations, or that there were no means by which to select. What might then have happened? In a sense, we know. For example, we know of the extreme vulnerability of populations when they encounter a disease that is new to them. And we know of the extreme vulnerability of overly "nice" communities when they become prey to "bad guys." What implications does this have for defining and measuring machine intelligence? Well, the answer would differ if we had in mind only specialist machines such as window washers, rather than colonizers of some hostile planet. However, for some purposes at least, I would think that what we would seek to define and measure—perhaps under the rubric of a generalized notion of "intelligence—would include attributes such as audacity, curiosity, and the ability to "mutate" (in a sense to be defined).

As a corollary, I would think that:

- Otimizing a mix of machines for complex autonomous tasks, the requirements for learning and adaptation would require that some of the machines be designed to experiment and "mutate," rather than always following what was thought by designers to be optimizing logic.

Much is known about dealing with uncertainty and much can be accomplished with sufficiently rigorous prior thinking about the circumstances that one may encounter. However, a principle in much of the work on uncertainty is that one cannot anticipate everything. Adaptation is essential. A great deal of adaptation can be accomplished if one has the appropriate building-block routines to try in different combinations. This said, some adaptations require new forms of organization, new processes, or both. One of the marvels of nature is self-organizing systems that have the "talent" to reorganize to deal with new circumstances.

6. SOME TECHNICAL ISSUES IN THINKING ABOUT BUILDING INTELLIGENT MODELS

Much has been written about artificial intelligence modeling. I would add here only a few observations based on personal experience. Some of this involved building a massive analytic war gaming system during the cold war, one in which we had Red, Blue, and Green agents representing the Soviet Union/Warsaw Pact, United States/NATO, and various third countries. These agents made decisions about war, strategy, escalation, deescalation, and termination amidst the events generated by a simulation [9].

The first observation is that such models may prove more useful if they reflect a strong design rather than, e.g., a more unstructured approach such as lots of miscellaneous rules and an inference engine. Even if performance in particular tasks might be high with the latter approach, credibility and understandability tend to go with structure and with the ability to trace rationales.

Machines will need models of other machines, and highly simplified models of the other machines' modeling. There is no infinite recursion here because—if for no other reason—uncertainties in key inputs to judgments are sufficiently large that fine-tuning doesn't work well. In our work, Blue's decisions were based on a model of Red, which in turn had a highly simplified model of Blue. Both Red and Blue could learn to some degree as the simulation proceeded, although this was wired-in learning such as changing planning factors based on events in the simulation and assessing which opponent model seemed best given observed behavior.

The second observation is that multiresolution modeling (MRM) is extremely important in such work (and in other types of modeling as well). By MRM I mean modeling that provides alternative levels at which to make inputs, as distinct from modeling that merely provides intermediate—and highly aggregated displays, but that does all calculations from the lowest level upward.[5]

MRM is important for many reasons, but one of them is relevant here. Higher level intelligent behavior depends on higher-level models, not on calculations from incredible depth. The reasons relate to the enormous uncertainties that exist at lower levels (higher resolution)—not only in "data," but also in algorithms. This is part of the celebrated "bounded rationality" problem explained by Herbert Simon. As a result, real people (and at least some intelligent models) must be able to reason and decide at the level of abstractions. Abstractions often get built into models willy-nilly, but there is great benefit in designing them in from the start. Ideally, models would also be able to infer their own abstractions on the basis of experience. That is surely plausible with newer technology, but we have a long way to go, to say the least. In the meantime, good design can be quite helpful. I believe that one of the best ways to "measure" the intelligence of machines will probably be to review the hierarchical concepts it uses and the processes used to move up and down those hierarchies. That is, just as we assess unintelligent computer programs not only in terms of sampled behavior, but also in terms of inputs, structure, etc., so also for intelligence.

Some of this has interesting linkages to common sense, understandability, cause-effect relationships, and learning. As a rule of thumb, I believe that a model intended to work at level n of resolution should be accompanied by models at levels n+1 and n-1. The more abstract version may be needed for planning functions such as screening, and the more detailed version may be needed to provide "explanations" (a highly relative concept) and the potential for a kind of learning that would adjust the level-n model. Experiences that may

[5] See [4],[10],[11],[12] for related discussion. Some authors use "multiresolution modeling" to mean only that a given model describes different objects or processes at different levels of resolution, without necessarily requiring that the model user can specify inputs at alternative levels. They sometimes fail to understand that truth does not always reside at the level of most detail or that much of our best information often comes at intermediate, or even low, levels of resolution.

appear magical at the intended level, level n, may be explainable at level n+1 of resolution and it may be possible to use the experiences to recalibrate lower-level assumptions and generate new abstractions. However, if the more detailed model doesn't even exist, then it would seem that the only recourse would be for the machine to use various and sundry techniques such as statistical analysis to infer what are additional variables. There are severe shortcomings to such an approach—if, at least, it is feasible to do better. This said, it is clear that humans do have the capability—with considerable effort—to see new things and find new ways to reason without them having been wired in our software. But we all know how useful it is to have analogies, metaphors, or theories to help.

It follows that one measure of intelligence might be the structural richness of reasoning models: is it sufficient to accommodate a good deal of experience-based learning?[6]

Merely as an example here, consider the choices one has in designing how to accomplish a disaggregation (i.e., moving from a relatively more to relatively lesser degree of abstraction, or from fewer to more variables). Economical reasoning, such as a maximum entropy principle, might suggest that the disaggregated view be as minimalist as possible given the explicit information available. However, one might instead want to consider disaggregations that "speculate" about much richer detailed structures. This, in a sense, is what happens in vision, where our mind "sees" far more than it has a right to see when viewed from a purely biological perspective: the mind draws on general knowledge to infer the presence of pàtterns. This process becomes evident primarily when the mind's first guess is wrong and we discontinuously shift to another one, perhaps noticing (if we try) that a shift is taking place.

Suppose, now, that our minds had been designed only to infer by an entropy maximizing principle. Our functionality would be substantially less. Instead, we have the ability to infer more than we see *and* the ability to do so iteratively, experimenting with different inferences, until the result makes sense. This is very different from what someone with a clockmaker's view of nature would have designed. This should tell us something about the design of intelligent machines.

7. EXPLORATORY ANALYSIS AS A KEY TO MEASURING INTELLIGENCE IN AN UNCERTAIN WORLD

One of the principles of our discussion of intelligence should be that the intelligence of a machine cannot usefully be judged independent of context. "Performance" measures exist, of course (e.g., processing speed), but

- How "intelligent" something is needs to be measured in relationship to both tasks to be done and contexts in which to do them.

These tasks and contexts, of course, may be extremely uncertain. This is obvious enough, but by analogy with my work in policy analysis I would argue that special methods are needed to make use of this notion. In particular, we should plan to construct what I have variously called "scenario spaces" or "assumptions spaces" in which to test our behaviors. Not only is it insufficient to pick an allegedly representative context, and work away at measurements for that, it is also not sufficient to do sensitivities around that context. Key reasons are as follows. First, there may not be a meaningful best-estimate or representative context; instead, there may be massive uncertainties that make any of many possible and very different contexts plausible. Second, the effects of contextual variables may be highly interactive, so that any linear approach to sensitivity testing would fail.

The approach my colleagues and I have used in this regard involves "exploratory analysis," which emphasizes studying the problem (e.g., assessing behavior's effectiveness) in a vast scenario space that is designed for comprehensiveness rather than detail. I refer to both parametric and probabilistic explorations. In the first, one discretizes the context's defining variables, and creates experimental designs that consider all (or a cleverly sampled subset) of the many combinations. In simple cases, we can do the full factorial design. In the second approach, one represents the defining variables' uncertainty with uncertainty distributions. Ultimately—after initial exploration—one settles on a hybrid approach in which some key variables are parameterized (so that one can see cause-effect relationships in output displays) and the others are treated probabilistically and convolved. This leads to a suggestion:

- Multiresolution, multiperspective exploratory analysis could be the basis for measuring the effectiveness of a machine over an enormous range of conditions (and with different measures). The results could be used to characterize intelligence—in multiple dimensions, and in different resolutions and perspectives, as appropriate.

Fortunately, recent technology makes a great deal of this type of thing feasible—even with PCs on our desktop at home. We are already at the stage where much can be learned by "flying through the space of outcomes" using clever graphics, and thereby seeing what regions (what combination of variable values) are most important (e.g., acceptable or unacceptable outcomes).

[6] Mystel [4] argues passionately for a multiresolution view of systems generally. As he emphasizes throughout, and as James Albus articulates in the preface, the point is not to have an MRM design so that one can move inexorably from the realm of detail to the level of abstractions in a pyramid-like structure, but to be able to move upward and downward at will—according to immediate needs.

These exploratory analysis methods could prove quite powerful in the task of assessing the intelligence of machines.

REFERENCES

[1] Gardner, Howard, *Frames of Mind: the Theory of Multiple Intelligences*, Basic Books, New York, 1983.

[2] Steinberg, Robert J., *The Triarchic Mind: a New Theory of Human Intelligence*, Viking, New York, 1988,

[3] National Institute of Standards and Technology, *Measuring Performance of Systems with Autonomy: a White Paper for Explaining Goals of the Workshop*, final draft, Alex Mystel, July, 2000.

[4] Meystel, Alex, *Semiotic Modeling and Situation Analysis: an Introduction*, AdRem Inc., Bala Cynwyd, PA, 1995.

[5] Keeney, Ralph and Howard Raiffa, *Decision with Multiple Objectives: References and Value Tradeoffs*, John Wiley and Sons, New York, 1976,

[6] Kleindorfer, Paul R., Howard C. Kunreuther, and Paul J. H. Schoemaker, Decision Sciences: An Integative Perspective, Cambridge Univeısity Press, New York, 1993.

[7] Hillestad, Richard and Paul K. Davis, *Resource Allocation for the New Defense Strategy: the DynaRank Decision Support System*, RAND MR-996, 1998.

[8] Klitgaard, Robert, *Choosing Elites: Selecting the Best and the Brightest at Top Universities and Elsewhere*, Basic Books, New York, 1985.

[9] Davis, Paul K. Davis, Paul K., *Some Lessons Learned from Building Red Agents in the RAND Strategy Assessment System (RSAS)*, RAND, N-3003-OSD. Available also in Military Operations Research Society, Mini-Symposium Proceedings: *Human Behavior and Performance as Essential Ingredients in Realistic Modeling of Combat–MORIMOC II*, 1989.

[10] Davis, Paul K. and James H. Bigelow, *Experiments in Multiresolution Modeliing*, RAND, 1998.

[11] Davis, Paul K., "Multiresoluton, Multiperspective Modeling (MRMPM) as an Enabler of Exploratory Analysis," *Proceedings of the SPIE*. Vol. 4026, 2000.

[12] National Research Council, *Post Cold War Conflict Deterrence*, National Academy Press, Washington, D.C., 1997.

Hierarchic Social Entropy and Behavioral Difference: New Measures of Robot Group Diversity[*]

Tucker Balch
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3891

Abstract

As research expands in multiagent intelligent systems, investigators need new tools for evaluating the artificial societies they study. It is impossible, for example, to correlate heterogeneity with performance in multiagent robotics without a quantitative metric of diversity. Currently diversity is evaluated on a bipolar scale with systems classified as either heterogeneous or homogeneous, depending on whether any of the agents differ. Unfortunately, this labeling doesn't tell us much about the extent of diversity in heterogeneous teams. How can it be determined if one system is more or less diverse than another? Heterogeneity must be evaluated on a continuous scale to enable substantive comparisons between systems. To enable these types of comparisons, we introduce: (1) a continuous measure of robot behavioral difference, and (2) hierarchic social entropy, an application of Shannon's information entropy metric to robotic groups that provides a continuous, quantitative measure of robot team diversity. The metric captures important components of the meaning of diversity, including the number and size of behavioral groups in a society and the extent to which agents differ. The utility of the metrics is demonstrated in the experimental evaluation of multirobot soccer and multirobot foraging teams.

1 Introduction

Heterogeneous systems are a growing focus of robotics research [FM97, GM97, Par94, Bal99]. Presently, diversity in these systems is evaluated on a bipolar scale; systems are classified as either *heterogeneous* or *homogeneous* depending on whether any of the agents differ. This view is limiting because it does not permit a quantitative comparison of heterogeneous systems. A principled study of diversity requires a quantitative metric. Such a metric would enable the investigation of issues like the impact of diversity on performance, and conversely, the impact of other task factors on diversity. To address this, we propose *social entropy* (computed using Shannon's information entropy formulation [Sha49]) as an appropriate measure of diversity in robot teams.

In this paper we briefly introduce the mathematical formulation of individual robot difference and robot societal diversity. More details and examples are provided in [Bal00].

2 The meaning of diversity

What does *diverse* mean? Webster [MW89] provides the following definition:

di.verse *adj* **1:** differing from one another: unlike **2:** composed of distinct or unlike elements or qualities

Clearly, difference plays a key role in the meaning of diversity. In fact, an important challenge in evaluating robot societal diversity is determining whether agents are alike or unlike. Assume for now that any two agents are either alike or not.

Now consider what *diverse* means for societies composed of several distinct subsets. To make the discussion more concrete, suppose the "society" under examination is a collection of four different shapes: circles, squares, triangles and stars. Figures 1 and 2 illustrate several sets of shapes as examples of ways the groupings can differ. The goal is to develop a quantitative metric that captures the meaning of diversity illustrated in these examples.

First, how should the number of distinct subsets in a society impact the measured diversity? Consider Figure 1: four sets of 12 shapes. Each set has a different number of homogeneous subsets; from one homogeneous subset in Figure 1a (all circles) to four in Figure 1d. This example suggests that the number of homogeneous subsets in a society is an important component of measured diversity.

[*]This is an abbreviated version an article published in *Autonomous Robots*, vol 8, no 3.

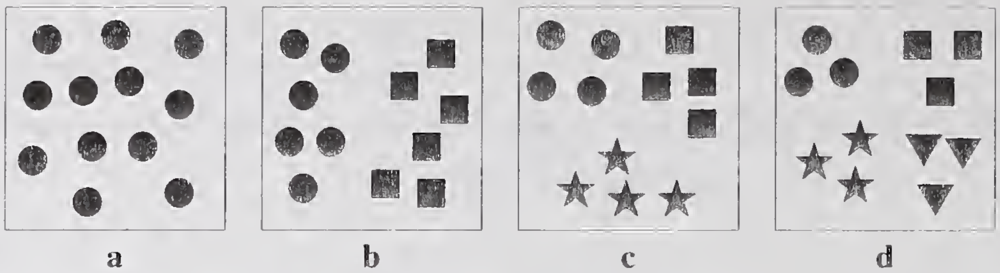


Figure 1. Several collections of shapes. The number of homogeneous subsets in each collection grows from one in a to four in d.

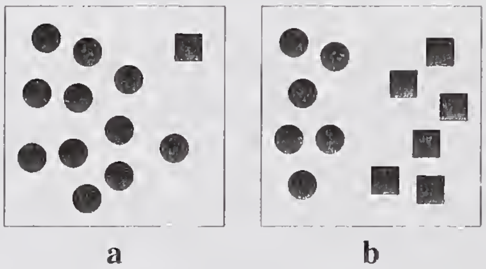


Figure 2. In both of these groups there are the same number of shapes and the same number of homogeneous subsets, but the proportion of elements in each subset is different.

Now consider Figure 2. Which group of shapes is more diverse? In both cases there are exactly 12 shapes and exactly two different types. In Figure 2a however, there is a much higher proportion of circles than in 2b where there is an equal number of circles and squares. This example suggests that the relative proportion of elements in each subset is an important component of diversity.

These examples highlight the fact that the *distribution* of the agents between homogeneous subsets is at the core of the meaning of diversity. In light of this observation, we make the following commitment: the measured diversity of a multiagent society depends on the number of homogeneous subsets it contains and the proportion of agents in each subset.

3 Simple social entropy

How should diversity be quantified? The properties Shannon sought in a measure of information uncertainty are also useful in the measurement of societal diversity [Sha49]. In fact, researchers in a number of disciplines have adopted information theoretic concepts of diversity. Information entropy is used by by ecologists as a means of evaluating species' diversity [LVW83, LW80, Mag88], by sociologists as a model of societal evolution [Bai90], and by taxonomists as a tool for evaluating classification methodologies [SS73, JS71].

Before proceeding we must introduce some notation:

- $\mathcal{R}$ is a society of N agents with $\mathcal{R} = \{r_1, r_2, r_3 ... r_N\}$
- $\mathcal{C}$ is a classification of $\mathcal{R}$ into M possibly overlapping subsets.
- c_i is an individual subset of $\mathcal{C}$ with $\mathcal{C} = \{c_1, c_2, c_3 ... c_M\}$
- $p_i = \frac{|c_i|}{\sum_{j=1}^{M} |c_j|}$ is the proportion of agents in the ith subset; and $\sum p_i = 1$.

In the last section we argued that the measured diversity of a system should reflect the number of groups in the system and the distribution of elements into those groups; diversity should therefore be a function of M and the p_is as defined above. Assume that a diversity metric exists and call it H. The diversity of a society partitioned into M homogeneous subsets is written $H(p_1, p_2, p_3, ..., p_M)$. So, for instance, the diversity of the group of shapes depicted in Figure 2a is $H\left(\frac{1}{12}, \frac{11}{12}\right)$, while the diversity for the group of shapes in Figure 2b is $H\left(\frac{1}{2}, \frac{1}{2}\right)$. The diversity of a particular robot society $\mathcal{R}_a$ can also be expressed $H(\mathcal{R}_a)$.

Shannon prescribed three properties for a measure of information uncertainty [Sha49]. With slight changes in notation, they are equally appropriate for a measure of societal diversity:

Property 1 continuous: H should be continuous in the p_i.

Property 2 monotonic: If all the p_i are equal, (i.e. $p_i = \frac{1}{M}$), then H should be a monotonically increasing function of M. In other words, if there are an equal number of agents in each subset, more subsets implies greater diversity.

Property 3 recursive: If a multiagent society is defined as the combination of several disjoint sub-societies, H for the new society should be the weighted sum of the individual values of H for the subsets. This property is important for the analysis of recursively composed societies (e.g. [MAC97]).

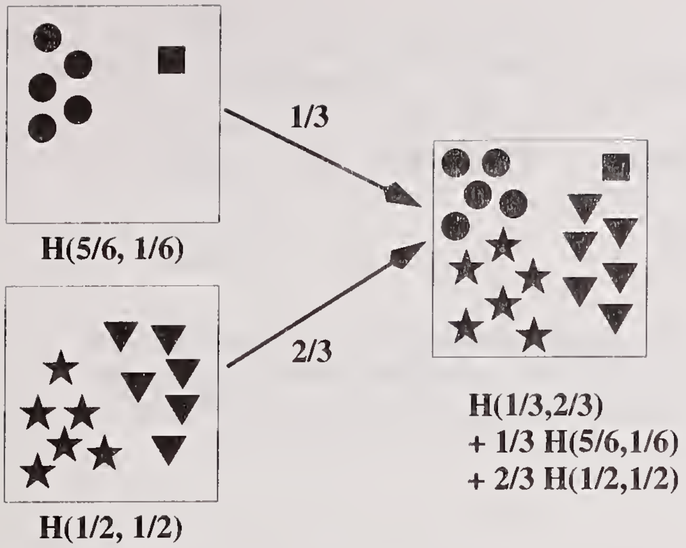


Figure 3. A new society (right) is generated by combining two others (left). The diversity of the new society is a weighted sum of the individual values of H for the subsets.

The meaning of the requirement that H be recursive is illustrated in Figure 3. The two groups on the left are combined into a new society on the right. In general, for a society $\mathcal{R}_c$ composed of two societies, $\mathcal{R}_a$ and $\mathcal{R}_b$, the recursive criteria ensures that:

$$H(\mathcal{R}_c) \quad = \quad H(\alpha, \beta) + \alpha H(\mathcal{R}_a) + \beta H(\mathcal{R}_b)$$

where α is the proportion of agents in $\mathcal{R}_a$, β is the proportion of agents in $\mathcal{R}_b$ and $\alpha + \beta = 1$.

Shannon's *information entropy* meets all three criteria [Sha49]. The information entropy of a random system X is given as[1]:

$$H(X) \quad = \quad -K \sum_{i=1}^{M} p_i \log_2(p_i) \qquad (1)$$

where K is a positive constant. Because K merely amounts to the choice of a unit of measure, Shannon sets $K = 1$ [Sha49]. Equation 1 (with $K = 1$) is adopted for the measurement of multiagent societal diversity. $H(\mathcal{R}_a)$ is the *simple social entropy* of agent society $\mathcal{R}_a$.

In addition to Properties 1, 2 and 3, H has a number of additional properties that further substantiate it as an appropriate measure of diversity. First, as we would expect, H is minimized for homogeneous societies; these groups are the least diverse. Also, for heterogeneous groups H is maximized when there are an equal number of agents in each subset. More precisely:

Property 4: $H = 0$ if and only if all the p_i but one are zero. In other words H is minimized when the system is homogeneous. Otherwise H is positive.

Property 5: For a given M (number of homogeneous subsets), H is maximized when all the p_i are equal, i.e. $p_i = \frac{1}{M}$. This is the case when there are an equal number of agents in each subset.

Property 6: Any change toward equalization of the proportions $p_1, p_2, \ldots, p_M$ increases H. Thus if $p_1 < p_2$ and we increase p_1, decreasing p_2 an equal amount so that they are more nearly equal, H increases. An important implication is that there are no locally isolated maxima.

Even if these properties are desirable in a diversity metric, why choose information entropy over another function possessing the same properties? Because, as it turns out, information entropy (Equation 1) is the *only* function satisfying Properties 1, 2 and 3. Shannon proved this result using the mathematically equivalent properties he required of an information uncertainty metric [Sha49].

The entropy of a number of example systems using this metric is given in Figure 4.

4 Classification and clustering

The discussion of diversity left open the question of how agents are classified into subsets. It was assumed that any two agents are either alike (in the same subset) or unlike. In actuality, the robotic agents to be classified are distributed in a multi-dimensional space where the dimensions correspond to components of behavior and difference corresponds to the distance between agents in the space. Difference between agents is likely to vary along a continuous spectrum instead of in the binary manner assumed previously.

The challenge of finding and characterizing clusters of elements distributed in a continuous multi-dimensional space is exactly the problem faced by biologists in building and using taxonomic systems. In the case of biology the dimensions of the space represent aspects of morphology or behavior that distinguish one organism from another. In this research the dimensions are the components of behavior that distinguish one robot from another.

[1] $H(X)$ is used in coding theory as a lower-bound on the average number of bits required per symbol to send multi-symbol messages. The random variable X assumes discrete values in the set $\{x_1, x_2, x_3 ... x_M\}$ (the alphabet to be encoded) and p_i represents the probability that $\{X = x_i\}$.

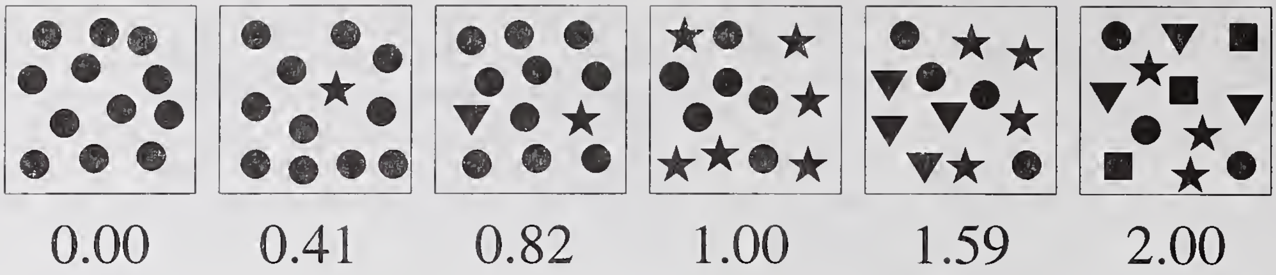


Figure 4. A spectrum of diversity. In the diagram above, each of the six squares encloses a multiagent system, from least diverse (homogeneous) on the left, to most diverse (most heterogeneous) on the right. The *simple social entropy*, **a qualitative measure of diversity, is listed underneath each system.**

The aims of taxonomic classification are distinct from other types of classification in that one goal is to arrange the elements in a hierarchy reflecting their distribution in the classification space. Conversely, many classification tasks only require a simple partitioning of the space (e.g. categorizing e-mail into folders). Taxonomic trees (the end result of the taxonomic classification process, e.g. Figure 5) are potentially more useful in the analysis of diversity than simple partitionings because they provide more information about the society's spatial structure.

Biology offers a rich literature addressing this problem. In fact, an entire field — *numerical taxonomy* — is devoted to ordering organisms hierarchically using principled numerical techniques [SS73, JS71]. Many of the approaches in numerical taxonomy are directly applicable to the problem of robot classification. They include mechanisms for building and analyzing classification structures (e.g. taxonomic trees) and for identifying organisms on the basis of these structures.

Techniques from numerical taxonomy address the problem of how to classify organisms, or groups of organisms, at various levels. At the lowest level in biological classification for instance, humans and gorillas are more likely to be classified together than, say, humans and dogs. But at a higher level, primates are in fact grouped with canines in the class *mammalia*. Dendrograms provide an orderly hierarchic view of the these classifications. While dendrograms *per se* are not necessary for the evaluation of diversity, they are useful visualization tools and their construction provides clues for the evaluation of overall societal diversity.

Dendrograms are constructed using a clustering algorithm parameterized by h, the maximum difference allowed between elements in the same subset. The notation $D(a, b)$ is used to refer to the difference between the elements a and b. In most applications the difference metric is normalized so that taxonomic distance between any two elements varies between 0 and 1. When $h = 1$ all elements are grouped together in one cluster (see the cluster at the top right in Figure 5 for example). As h is reduced from 1 down to 0

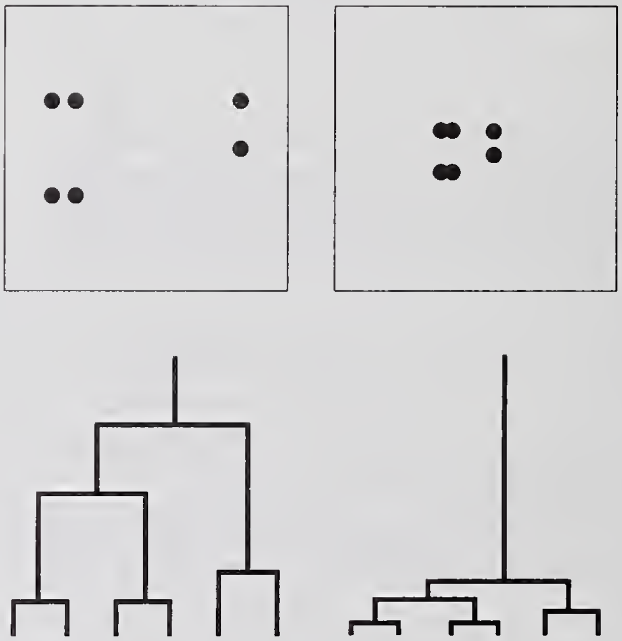


Figure 6. The branching structure of the dendrograms for these two societies is the same. However, the more compact distribution of elements in the system on the upper right is reflected in the branches being compressed towards the bottom of the corresponding dendrogram (lower right).

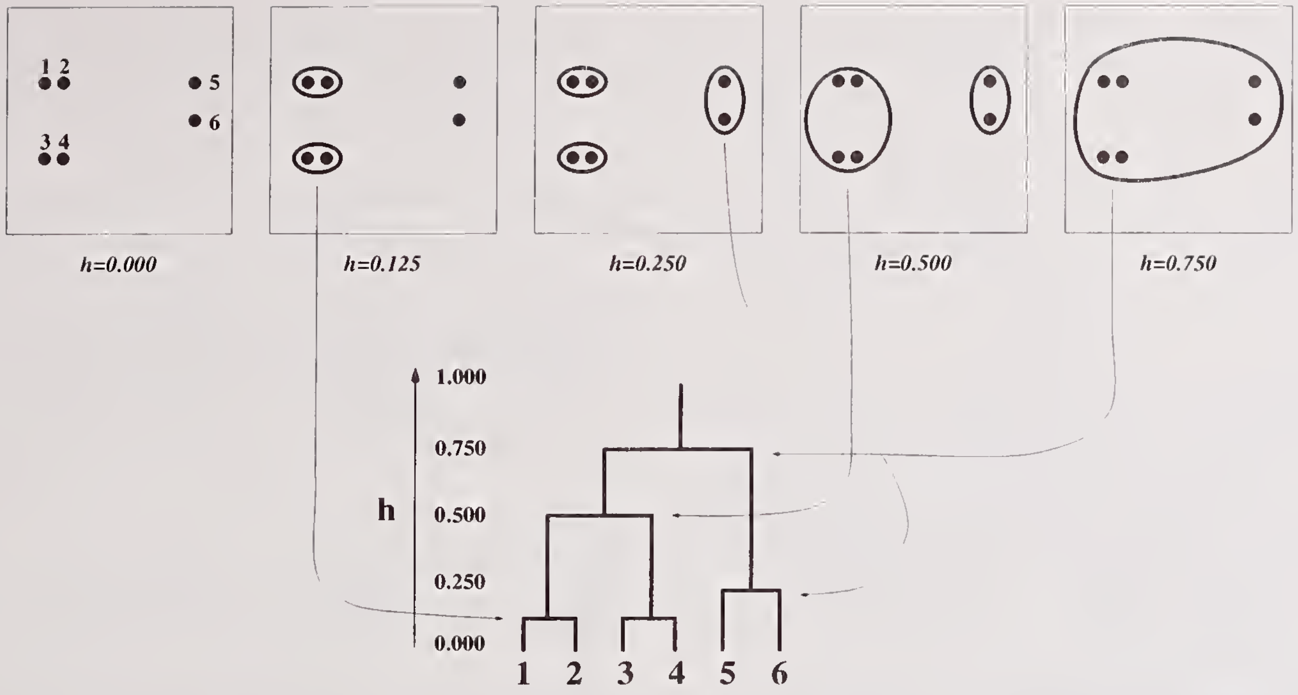


Figure 5. Example classification using numerical techniques. The top row shows how the system is clustered at several levels, parameterized by taxonomic level h (h is distinct from information entropy H). The classification is summarized in a taxonomic tree, or dendrogram (bottom). Strong similarities between elements are indicated by grouping near the bottom of the dendrogram; weaker similarities between groups are reflected in converging branches at higher levels.

cluster boundaries change; the number of subsets increases as they split into smaller clusters. The splits are reflected as branches in the dendrogram. Finally, when $h = 0$ each element is a separate cluster; a "leaf" at the bottom of the dendrogram "tree."

Dendrograms can reveal subtle differences in societal structure. Figure 6 for example, shows two societies with the same relative arrangement of elements, but one grouping is compact while the other is spread out over a larger area. The difference in scale is reflected in a compressed dendrogram for the spatially compact society (Figure 6 right). Can these differences be accounted for in the evaluation of diversity?

The spatial extent of elements in a taxonomic space is a reflection of the degree of difference between agents. Note that sensitivity to the degree of difference between elements in hierarchic clustering depends on h. Because h is a *parameter* of the clustering algorithm, it can be varied to examine clusterings at any scale. Hierarchic algorithms are, in effect, variable power clustering microscopes. For values of h near zero the tiniest difference between elements will cause them to be classified separately, while the clusterings at large values of h reveal societal structure at a macroscopic level. This feature is exploited in the development of a diversity measure sensitive to differences in the spatial size of societies.

5 Hierarchic social entropy

Now consider how these tools from numerical taxonomy can be applied to the measurement of diversity. The discussion of hierarchic clustering algorithms above described how the number and size of clusters depend on h. But how is simple social entropy impacted by changes in h? Since the partitioning of a society is based on h the entropy also depends on it. An example of the relationship is illustrated in Figure 7. Entropy changes in discrete steps as h increases. Note that points where change occurs correspond to branch points in the dendrogram.

Compare the dendrograms and entropy plots of the two societies in Figure 7. As in the earlier example, the two groups have the same relative structure, but the society represented on the right is more compact, resulting in branching compressed towards the bottom of the tree. The difference in scale is also readily apparent in the plots of entropy. Entropy drops to zero much more quickly in the plot corresponding to the compact society. Because the value of simple entropy depends significantly on h when hierarchic clustering is used, we augment the notation to account for this:

$$H(\mathcal{R}, h) = H(\mathcal{R}) \text{ for the clustering of } \mathcal{R} \text{ at taxonomic level} \quad (2)$$

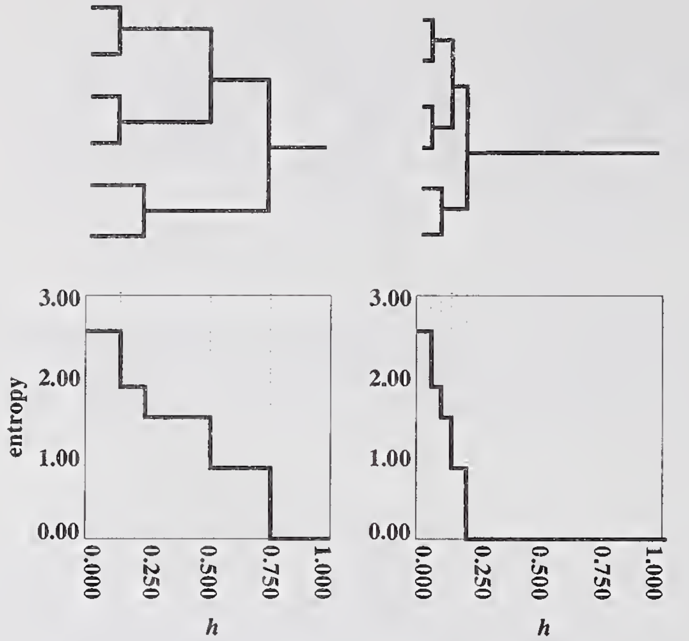


Figure 7. Entropy depends on h. A comparison of entropy versus h for for two societies. For clarity, the dendrogram is rotated 90 degrees.

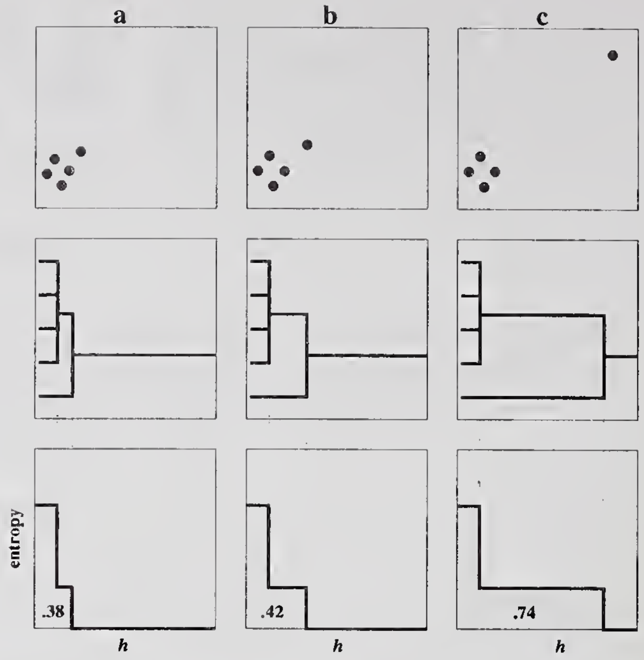


Figure 8. Hierarchic social entropy (bottom) is computed for three societies (top). The values are 0.715 for the system on the left and 1.00 for the system on the right. The calculated value increases as the element on the upper right is positioned further away from the group. Dendrograms for the groups are also displayed (middle row).

H is a function of $\mathcal{R}$ *and* h because the classification of agents into subsets, and therefore the entropy, depends on them both. This highlights the fact that the entropy of a particular clustering is only a snapshot of the society's diversity. A comprehensive evaluation of diversity should account for clustering at all taxonomic levels. This is easily accomplished using the area under the entropy plot as a measure of diversity. This augmented metric, called *hierarchic social entropy*, is defined as:

$$S(\mathcal{R}) = \int_0^\infty H(\mathcal{R},h)dh \qquad (3)$$

where $\mathcal{R}$ is the robot society under evaluation, h is a parameter of the clustering algorithm indicating the maximum difference between any two agents in the same group and $H(\mathcal{R},h)$ is the simple entropy of the society for the clustering at level h. Note that as $h \rightarrow \infty$ a point is reached where all elements are clustered in the same subset (the maximum taxonomic distance). $H(\mathcal{R},h)$ drops to 0 at this point. In the behavioral difference measure used in this work, the maximum possible difference between elements is fixed at 1.0, so the upper limit of the integration is 1 rather than ∞ as in the general case.

Hierarchic social entropy is a continuous ratio measure; it has an absolute zero (when all elements are identical) and equal units. This enables a total ordering of societies on the basis of diversity. It also provides for quantitative results of the form "$\mathcal{R}_b$ is *twice* as diverse as $\mathcal{R}_a$." This is a significant advantage over the categorization of systems as simply "homogeneous" or "heterogeneous." Three example calculations of hierarchic social entropy are provided in Figure 8.

6 Behavioral difference

To summarize: hierarchic clustering is a means of dividing a society into subsets of behaviorally equivalent agents at a particular taxonomic level. Diversity is evaluated at each taxonomic level based on the number of subsets and the number of robots in each subset at that level. Integrating the diversity across all taxonomic levels produces an overall measure of diversity for the system. Previous sections have described the overall diversity metric and algorithms for clustering the agents into subsets. This section focuses on the difference metric used for clustering.

How should the behavior of two agents be compared? The technique advocated here is to look for differences in the agents' behavioral coding. In many cases (e.g. [BBC$^+$95, Mat92, GM97]) robot behavior is coded statically ahead of time, thus individuals may be directly compared by evaluating their behavioral configuration. Learning multirobot systems (e.g. [Bal97, Mat94]) pose a challenge because their behavior evolves over time. To avoid that problem in this research, the policies of learning agents are evaluated after agents converge to stable behavior.

This approach depends on three key assumptions:

Assumption 1: At the time of comparison, the robots' policies are fixed and deterministic.

Assumption 2: The robots under evaluation are substantially mechanically similar: differences in overt behavior are influenced more significantly by differences in policy than by differences in hardware.

Assumption 3: Differences in policy are correlated with differences in overt behavior.

If these conditions are not met in a particular multirobot system, the approach may not be appropriate. But the assumptions are reasonable for the conditions of this research, namely: experiments conducted on mechanically similar robots built on the same assembly line. Control systems running on the robots differ only in the data specifying each agent's policy. The comparison of these policies is the crux of the approach.

To facilitate the discussion, the following additional symbols and terms are defined:

- i is a robot's perceptual state.
- a is the action (behavioral assemblage) selected by a robot's control system based on the input i.
- π_j is r_j's policy; $a = \pi_j(i)$.
- p^i_j is the number of times r_j has encountered perceptual state i divided by the total number of times all states have been encountered. Experimentally, p^i_j is computed *post facto.*

The approach is to evaluate behavioral difference by comparing the robots' policies. The two foraging robots introduced earlier, for example, exhibit behavioral differences that are reflected in and caused by their differing policies. In the terminology introduced above, i represents the perceptual features an agent uses to selectively activate behaviors.

Definition 1: r_a and r_b, are **absolutely behaviorally equivalent** iff they select the same behavior in every perceptual state.

In complex systems with perhaps thousands of states and hundreds of actions it may also be useful to provide a scale of equivalence. This would allow substantially similar agents to be grouped in the same cluster even though they differ by a small amount. The approach is to compare two robots, r_a and r_b, by integrating the differences between their responses, $\mid \pi_a(i) - \pi_b(i) \mid$ over all perceptual states i. If the action is a single-dimension scalar, as in a motor current for instance, the difference can be taken directly. However, complex actions like *wander* and *acquire* are treated as nominal values with response difference defined as 0 when $\pi_a(i) = \pi_b(i)$ and 1 otherwise. This approach is often used in classification applications to quantify difference between nominal variables (e.g. eye color, presence or absence of a tail, etc.). Using this notation, a simple behavioral difference metric can be defined as:

$$D'(r_a, r_b) \quad = \quad \frac{1}{n} \int \mid \pi_a(i) - \pi_b(i) \mid di \qquad (4)$$

or for discrete state/action spaces:

$$D'(r_a, r_b) \quad = \quad \frac{1}{n} \sum_i \mid \pi_a(i) - \pi_b(i) \mid \qquad (5)$$

where $\frac{1}{n}$ is a normalization factor to ensure the difference ranges from 0 to 1. In the case of the discrete sum, n corresponds to the number of possible states. If r_a and r_b select identical outputs ($\pi_a(i) = \pi_b(i)$) in all perceptual states (i), then $D'(r_a, r_b) = 0$. When r_a and r_b select different outputs in all cases $D'(r_a, r_b) = 1$. In the numerical taxonomy literature, this difference is called the *mean character difference* [SS73]. The calculation parallels the idealized evaluation chamber procedure introduced earlier (Figure **??**).

Equations 4 and 5 weigh differences equally across all perceptual states. This may be problematic for agents that spend large portions of their time in a small portion of the states. Consider two foraging robots that differ only in their reaction to blue attractors. If, in their environment, no blue attractors are present the agents would appear to an observer to have identical policies.

There may be other important reasons certain states are never visited. In learning a policy, for instance, the robots might discover in early trials that certain portions of the state space should be avoided due to large negative rewards. Because these portions of the space are avoided, the agents will not refine their policies there, but avoid them entirely. It is entirely possible for the agents to differ significantly in these portions of the space even though they may appear externally to behave identically.

To address this, the response differences in states most frequently visited should be emphasized while those that are infrequently experienced should be de-emphasized. This is accomplished by multiplying the response difference in each situation by the proportion of times that state was visited by each agent $(p^i_a + p^i_b)$. Formally, **behavioral difference** between two robots r_a and r_b is defined as:

$$D(r_a, r_b) \quad = \quad \int \frac{(p^i_a + p^i_b)}{2} \mid \pi_a(i) - \pi_b(i) \mid di \qquad (6)$$

or in discrete spaces

$$D(r_a, r_b) \quad = \quad \sum_i \frac{(p^i_a + p^i_b)}{2} \mid \pi_a(i) - \pi_b(i) \mid \qquad (7)$$

When r_a and r_b select differing outputs in a given situation, the difference is normalized by the joint proportion of times they have experienced that situation.

7 Conclusion

This work is motivated by the idea that behavioral diversity should be evaluated as a *result* rather than an initial condition of multirobot experiments. Previously, researchers configured robot teams as homogeneous or heterogeneous *a priori*, then compared performance of the resulting teams [FM97, GM97, Par94]. That approach does not support the study of behavioral diversity as an emergent property in multirobot teams.

Defining behavioral diversity as an independent rather than dependent variable enables the examination of heterogeneity from an ecological point of view. How and when does diversity arise in robot teams interacting with each other and their environment? This work provides the necessary quantitative measures for this new type of investigation.

In this paper we introduce a mathematical definition of agent difference that can be used to group agents according to similarity. The grouping (or clustering) of agents is parameterized by h, a limit on how different agents can be, yet still be grouped in the same cluster. An overall diversity metric, hierarchical social entropy may then be computed using the difference metric, h, and clustering algorithms originally developed by biologists for taxonomic classification.

References

[Bai90] K. Bailey. *Social Entropy Theory*. State University of New York Press, Albany, 1990.

[Bal97] Tucker Balch. Learning roles: Behavioral diversity in robot teams. In *AAAI-97 Workshop on Multiagent Learning*, Providence, R.I., 1997. AAAI.

[Bal99] T. Balch. The impact of diversity on performance in multirobot foraging. In *Proc. Autonomous Agents 99*, Seattle, WA, 1999.

[Bal00] T. Balch. Hierarchic social entropy: An information theoretic measure of robot group diversity. *Autonomous Robots*, 8(3), July 2000. to appear.

[BBC+95] T. Balch, G. Boone, T. Collins, H. Forbes, D. MacKenzie, and J. Santamaría. Io, Ganymede and Callisto - a multiagent robot trash-collecting team. *AI Magazine*, 16(2):39–51, 1995.

[Dem92] L. Demetrius. The thermodynamics of evolution. *Physica A*, 189(3-4):417–436, November 1992.

[FM97] M. Fontan and M. Mataric. A study of territoriality: The role of critical mass in adaptive task division. In *From Animals to Animats 4: Proceedings of the Fourth International Conference of Simulation of Adaptive Behavior*, pages 553–561. MIT Press, 1997.

[GM97] D. Goldberg and M. Mataric. Interference as a tool for designing and evaluating multi-robot controllers. In *Proceedings, AAAI-97*, pages 637–642, July 1997.

[JS71] N. Jardine and R. Sibson. *Mathematical Taxonomy*. John Wiley & Sons, 1971.

[LVW83] D. Lurie, J. Valls, and J. Wagensberg. Thermodynamic approach to biomass distribution in ecological systems. *Bulletin of Mathematical Biology*, 45(5):869–872, 1983.

[LW80] D. Lurie and J. Wagensberg. Information theory and ecological diversity. In L. Garrido, editor, *Systems Far from Equilibrium*, pages 290–303, Berlin, West Germany, 1980. Sitges Conf. on Statistical Mechanics, Springer-Verlag.

[MAC97] D. MacKenzie, R. Arkin, and J. Cameron. Multiagent mission specification and execution. *Autonomous Robots*, 4(1):29–52, 1997.

[Mag88] A.E. Magurran. *Ecological Diversity and Its Measurement*. Princeton University Press, 1988.

[Mat92] M. Mataric. Designing emergent behaviors: From local interactions to collective intelligence. In *Proceedings of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats 2*, pages 432–441, 1992.

[Mat94] M. Mataric. Learning to behave socially. In *Proceedings of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats 3*, 1994.

[MW89] Merriam-Webster. *Webster's ninth new collegiate dictionary*. Merriam-Webster, 1989.

[Par94] Lynne E. Parker. *Heterogeneous Multi-Robot Cooperation*. PhD thesis, M.I.T. Department of Electrical Engineering and Computer Science, 1994.

[Sha49] C. E. Shannon. *The Mathematical Theory of Communication*. University of Illinois Press, 1949.

[SS73] P. Sneath and R. Sokal. *Numerical Taxonomy*. W. H. Freeman and Company, San Francisco, 1973.

[Wil92] E.O. Wilson. *The Diversity of Life*. Norton, 1992.

The Developmental Approach to Evaluating Artificial Intelligence—A Proposal

Anat Treister-Goren* and Jason Hutchens
Artificial Intelligence NV

ABSTRACT

We propose a developmental evaluation procedure for artificial intelligence[1] that is based on two assumptions: that the Turing Test provides a sufficient subjective measure of artificial intelligence, and that any system capable of passing the Turing Test will necessarily incorporate behavioristic learning techniques.

KEYWORDS: *artificial intelligence, human-computer conversation, Turing Test, child machine, verbal behavior, Markov modeling, information theory*

1. INTRODUCTION

In 1950 Alan Turing considered the question "Can machines think?" Turing's answer to this question was to define the meaning of the term 'think' in terms of a conversational scenario, whereby if an interrogator cannot reliably distinguish between a machine and a human based solely on their conversational ability, then the machine could be said to be thinking [1]. Originally called the imitation game, this procedure is nowadays referred to as the Turing Test.

The field of artificial intelligence (AI) has largely ignored this strict evaluation criterion. Today AI encompasses topics such as intelligent agents, chatterbots, pattern recognition systems, voice recognition systems and expert systems, with applications in medicine, finance, entertainment, business and manufacturing. It could be said that the field is currently in a contentious state. Even though important work has been conducted in terms of the sophistication and expertise of programs, the vision which motivated the birth of AI has not yet been fulfilled: there is neither sufficient cooperation nor agreement amongst its researchers. The unfortunate result of this trend is that true advancement is inhibited. We believe that a new approach is required.

In this paper we shall demonstrate that the Turing Test is a sufficient evaluation criteria for artificial intelligence provided that the expectation level of the interrogator is set appropriately. We propose to achieve this by complementing the Turing Test with objective developmental evaluation. The logical flow of this paper reflects the necessary steps one must take when trying to establish evaluation standards for artificial intelligence: we begin with a definition of artificial intelligence, we continue with a discussion of the theory and methods which we believe are an essential prerequisite for the emergence of artificial intelligence and we conclude with our proposed evaluation procedure.

2. THE TURING TEST

The Turing Test is an appealing measure of artificial intelligence because, as Turing himself writes, it ...

> ... has the advantage of drawing a fairly sharp line between the physical and the intellectual capacities of a man.

The Loebner Contest, held annually since 1991, is an instantiation of the Turing Test [2]. The sophistication and performance of computer programs entered into the contest, or lack thereof, bears out our introductory remark that the Turing Test has been largely ignored by the field. In a recent thorough review of conversational systems, Hasida and Den emphasize the absurdity of performance in the Loebner Contest [3]. They assert that since the Turing Test requires that systems "talk like people", and since no system currently meets this requirement, the *ad-hoc* techniques which the Loebner Contest subsequently encourages make little contribution to the advancement of dialog technology.

Although we agree wholeheartedly that the Loebner Contest has failed to contribute to the advancement of artificial intelligence, we do believe that the Turing Test is an appropriate evaluation criteria, and therefore our approach equates artificial intelligence with conversational skills. We further believe that engaging in domain-unrestricted conversation is the most critical evidence of intelligence.

2.1. Turing's Child Machine

Turing concluded his classic paper by theorizing on the design of a computer program which would be capable of passing the Turing Test. He correctly anticipated the limitations of simulating adult level conversation, and proposed that ...

> ... instead of trying to produce a program to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain.

*All correspondence should be emailed to anat@a-i.com.

[1] We use the term *artificial intelligence* to refer to machine intelligence which exhibits human-like conversational capability. To avoid ambiguity, we shall refer to the field of artificial intelligence as AI throughout.

Turing regarded language as an acquired skill, and recognized the importance of avoiding the hard-wiring of the computer program wherever possible. He viewed language learning in a behavioristic light, and believed that the language channel, narrow though it may be, is sufficient to transmit the information which the child machine requires in order to acquire language.

It is indeed unfortunate that this promising line of work was mostly abandoned by the field. Today we find ourselves at a crossroads—a paradigm shift is in the air, and many AI researchers are returning to the behavioristic approach that Turing favoured.

2.2. The Traditional Approach

Contrary to Turing's prediction that at about the turn of the millennium computer programs will participate in the Turing Test so effectively that an average interrogator will have no more than a seventy percent chance of making the right identification after five minutes of questioning, no true conversational systems have yet been produced, and none has passed an unrestricted Turing Test.

This may be due in part to the fact that Turing's idea of the child machine has remained unexplored—the traditional approach to conversational system design has been to equate language with knowledge, and to hard-wire rules for the generation of conversations. This approach has failed to produce anything more sophisticated than domain-restricted dialog systems which lack the kind of flexibility, openness and capacity to learn that are the very essence of human intelligence. As far as human-like conversational skills are concerned, no system has surpassed toddler level, if at all.

Since the 1950's, the field of child language research has undergone a revolution, inspired by Chomsky's transformational grammar [4] on the one hand and Skinner's behaviorist theory of language [5] on the other. Computational implementations based on the Chomskian philosophy are the norm, and have yielded disappointing results. It is our thesis that true conversational abilities are more easily obtainable via the currently neglected behavioristic approach.

3. VERBAL BEHAVIOR

Behaviorism focuses on the observable and measurable aspects of behavior. Behaviorists search for observable environmental conditions, known as *stimuli*, that co-occur with and predict the appearance of specific behavior, known as *responses* [6]. This is not to say that behaviorists deny the existence of internal mechanisms; they do recognize that studying the physiological basis is necessary for a better understanding of behavior. What behaviorists object to are internal structures or processes with no specific physical correlate inferred from behavior.

Behaviorists therefore object to the kind of grammatical structures proposed by linguists, claiming that these only complicate explanations of language acquisition [7]. They favour a functional rather than a structural approach, focusing on the function of language, the stimuli that evoke verbal behavior and the consequences of language performance. We believe this to be the right approach for the generation of artificial intelligence.

Skinner argues that psycholinguists should ignore traditional categories of linguistic units, and should instead treat language as they would any other behavior. That is, they should search for the functional units as they naturally occur, and then discover the functional relationship that predict their occurrence.

Behaviorism focuses on reinforced training. Since language is regarded as a skill that is not essentially different from any other behavior, generating and understanding speech must therefore be controlled by stimuli from the environment in the form of reinforcement, imitation and successive approximations to mature performance. Skinner takes the extreme position that the speaker is merely a passive recipient of environmental pressures, having no active role in the process of language behavior or development.

According to behaviorists, changes in behavior are explained through the association of stimuli in the environment with certain responses of the organism. The processes of forming such associations are known as *classical conditioning* and *operant conditioning.*

3.1. Classical Conditioning

Classical conditioning accounts for the associations formed between arbitrary verbal stimuli and internal responses or reflexive behavior. In classical conditioning, for example, the word 'milk' is learned when the infant's mother says "milk" before or after feeding, and this word becomes associated with the primary stimulus (the milk itself) to eventually elicit a response similar to the response to the milk. Once a word or a *conditioned stimulus* elicits a *conditioned response*, it can become an *unconditioned stimulus* for modifying the response to another conditioned stimulus. For example, if the new conditioned stimulus 'bottle' frequently occurs with the word 'milk', it may come to elicit a response similar to that for the word 'milk'. Words stimulate each other and classical conditioning accounts for the interrelationship of words and word meanings. Classical conditioning is more often used to account for the receptive side of language acquisition.

3.2. Operant Conditioning

Operant conditioning is used to account for changes in voluntary, non-reflexive behavior that arise due to environmental consequences contingent upon that behavior. All behavioristic accounts of language acquisition assume that children's productive speech develops through differential reinforcers and punishers supplied by environmental agents in a process known as *shaping.* Children's speech that most closely resembles adult speech is rewarded, whereas productions that are meaningless are either ignored or punished. Behaviorists believe that the course of language development is largely determined by the course of training, not maturation, and that the time it takes children to acquire language is a consequence of the limitations of the training techniques. Operant conditioning is used to account for the productive side of language acquisition.

Imitation is another important factor in language acquisition because it allows the laborious shaping of each and every verbal response to be avoided. The process of imitation itself becomes reinforcing and enables rapid learning of complex behaviors.

Behaviorists do not typically credit the child with intentions or meanings, the knowledge of rules or the ability to abstract important properties from the language of the environment. Rather, certain stimuli evoke and strengthen certain responses in the child. The sequence of language acquisition is determined by the most salient environmental stimuli at any point in time, and by the child's past experience with those stimuli. The learning principle of reinforcement is therefore taken to play a major role in the process of language acquisition, and is the one we believe should be used in creating artificial intelligence.

4. THE DEVELOPMENTAL MODEL

We maintain that a behavioristic developmental approach could yield breakthrough results in the creation of artificial intelligence. Programs can be granted the capacity to imitate, to extract implicit rules and to learn from experience, and can be instilled with a drive to constantly improve their performance. Language acquisition can be achieved through successive approximations and positive and negative feedback from the environment. Once given these capabilities, programs should be able to evolve through critical developmental language acquisition milestones in order to reach adult conversational ability.

Human language acquisition milestones are both quantifiable and descriptive, and any system that aims to be conversational can be evaluated as to its analogical human chronological age. Such systems could therefore be assigned an age or a maturity level beside their binary Turing Test assessment of "intelligent" or "not intelligent".

4.1. Success in Other Fields

Developmental principles have enabled evaluation and treatment programs in fields formerly suffering from a lack of organizational and evaluative principles [8], [9], and have been especially useful in areas which border on the question of intelligence. Normative developmental language data has enabled the establishment of diagnostic scales, evaluation criteria and treatment programs for developmentally delayed populations. In other areas, such as schizophrenic thought disorder, in which clinicians often found themselves unable to capture the communicative problem of patients in order to assess their intelligence level or cognitive capability, let alone to decipher medication treatment effects, the developmental approach has proven to be a powerful tool [10].

5. LANGUAGE MODELING

We are interested in programming a computer to acquire and use language in a way analogous to the behavioristic theory of child language acquisition. In fact, we believe that fairly general information processing mechanisms may aid the acquisition of language by allowing a simple language model, such as a low–order Markov model, to bootstrap itself with higher-level structure.

5.1. Markov Modeling

Claude Shannon, the father of Information Theory, was generating quasi-English text using Markov models in the late 1940's [11]. Such models are able to predict which words are likely to follow a given finite context of words, and this prediction is based on a statistical analysis of observed text. Using Markov models as part of a computational language acquisition system allows us to minimize the number of assumptions we make about the language itself, and to eradicate language-specific hard-wiring of rules and knowledge.

Some behaviorists explain that language is processed as word-sequences, or response-chains, with the words themselves serving as stimulus for their successors [12]. Information theoretic measures may be applied to Markov models to yield analogous behavior, and more sophisticated techniques can model the case where long-distance dependencies exist between the stimulus and the response.

To date, conversation systems based on this approach have been thin on the ground [13], although the technique has been used extensively in related problems, such as speech recognition, text disambiguation and data compression [14].

5.2. Finding Higher-Level Structure

Information theoretic measures may be applied to the predictions made by a Markov model in order to find sequences of symbols and classes of symbols which constitute higher-level structure. For example, in the complete absence of *a priori* knowledge of the language under investigation, a character-level Markov model inferred from English text can easily segment the text into words, while a word-level Markov model inferred from English text may be used to 'discover' syntactic categories [15].

This structure, once found, can be used to bootstrap the Markov model, allowing it to capture structure at even higher levels. A hierarchy of models is thus formed, each of which views the data at a different level of abstraction. Although each level of the hierarchy is formed in a purely bottom-up fashion from the data supplied to it by the level below, the fact that each model provides a top-down view with respect to the models below it allows a feedback process to be applied, whereby interaction between models at adjacent levels of abstraction serves to correct bad generalisations made in the bootstrapping phase.

It is our belief that combining this approach with positive and negative reinforcement is a sensible way of realizing Turing's vision of a child machine.

6. EVALUATION PROCEDURE

Our proposal is to measure the performance of conversational systems via both subjective methods and objective developmental metrics.[2]

6.1. Objective Developmental Metrics

The ability to converse is complex, continuous and incremental in nature, and thus we propose to complement our subjective impression of intelligence with objective incremental metrics. Examples of such metrics, which increase quantitatively with age, are:

Vocabulary size: The number of different words spoken.

Mean length of utterance: The mean number of word morphemes spoken per utterance.

Response types: The ability to provide an appropriate sentence form with relevant content in a given conversational context, and the variety of forms used.

Degree of syntactic complexity: For example, the ability to use embedding to make connections between sentences, and to convey ideas.

The use of pronominal and referential forms: The ability to use pronouns and referents appropriately and meaningfully.

These metrics provide an evaluation of progress in conversational capability, with each capturing a specific aspect. Together they enable an understanding of the nature of the critical abilities that contribute toward our desired goal: achieving a subjective judgement of intelligence.

The challenge in creating maturational criteria is in combining these metrics meaningfully. One might expect discrepancies in the development of the different aspects of conversational performance. For example, some systems may utter long, syntactically complex sentences, typical of a child aged five or above, but may lag in terms of the use of pronouns expected at that age. Weighting the various developmental metrics is far from trivial.

6.2. The Subjective Component

We do not claim that objective evaluation should take precedence over subjective evaluation, just as we do not judge children on the basis of objective measures alone. Subjective judgement is an important if not determining criterion of overall evaluation. We believe that the subjective evaluation of artificial intelligence is best performed within the framework of the Turing Test.

The judgement of intelligence is in the eye of the beholder. Human perception of intelligence is always influenced by the expectation level of the judge toward the person or entity under scrutiny—obviously, intelligence in monkeys, children or university professors will be judged differently. Using objective metrics to evaluate maturity level will help set up the right expectation level to enable a valid subjective judgement to be made.

Accordingly, we propose that suitable developmental metrics be chosen in order to establish a common denominator among various conversational systems so that the expectation level of these systems will be realistic. Given that subjective impression is at the heart of the perception of intelligence, the constant feedback from the subjective evaluation to the objective one will eventually contribute to an optimal evaluation system for perceiving intelligence.

By using the developmental model, computer programs will be evaluated to have a maturity level in relation to their conversational capability. Programs could be at the level of toddlers, children, adolescents or adults depending on their developmental assessment. This approach enables evaluation not only across programs but also within a given program.

7. CONCLUSION

We submit that a developmental approach is a prerequisite to the emergence of intelligent lingual behavior and to the assessment thereof. This approach will help establish standards that are in line with Turing's understanding of intelligence, and will enable evaluation across systems.

We predict that the current paradigm shift in understanding the concepts of AI and natural language will result in the development of groundbreaking technologies which will pass the Turing Test within the next ten years.

8. REFERENCES

[1] A.M. Turing, "Computing machinery and intelligence," in *Collected works of A.M. Turing: Mechanical Intelligence*, D.C. Ince, Ed., chapter 5, pp. 133–160. Elsevier Science Publishers, 1992.

[2] Stuart M. Shieber, "Lessons from a restricted Turing test," Available at the Computation and Language e-print server as `cmp-lg/9404002`., 1994.

[3] K. Hasida and Y. Den, "A synthetic evaluation of dialogue systems," in *Machine Conversations*, Yorick Wilks, Ed. Kluwer Academic Publishers, 1999.

[4] Noam Chomsky, *Syntactic Structures*, Mouton, 1975.

[2] We use the term *metric* in its non-mathematical sense of relating to measurement.

[5] B.F. Skinner, *Verbal Behavior*, Prentice-Hall, 1957.

[6] R.E. Owens, *Language Development*, Macmillan Publishing Company, 1992.

[7] G. Whitehurst and B. Zimmerman, "Structure and function: A comparison of two views of development of language and cognition," in *The Functions of Language and Cognition*, G. Whitehurst and B. Zimmerman, Eds. Academic Press, 1979.

[8] J.B. Gleason, *The Development of Language*, Charles E. Merrill Publishing Company, 1985.

[9] A. Goren, G. Tucker, and G.M. Ginsberg, "Language dysfunction in schizophrenia," *European Journal of Disorders of Communication*, vol. 31, no. 2, pp. 467–482, 1996.

[10] A. Goren, "The language deficit in schizophrenia from a developmental perspective," in *The Israeli Association of Speech and Hearing Clinicians*, 1997.

[11] Claude E. Shannon and Warren Weaver, *The Mathematical theory of Communication*, University of Illinois Press, 1949.

[12] O.H. Mowrer, *Learning Theory and Symbolic Processes*, Wiley, 1960.

[13] Jason L. Hutchens, "Introducing MegaHAL," in *NeMLaP3 / CoNLL98 Workshop on Human-Computer Conversation, ACL*, David M. W. Powers, Ed., January 1998, pp. 271–274.

[14] Eugene Charniak, *Statistical Language Learning*, MIT Press, 1993.

[15] Jason L. Hutchens, "Finding structure via compression," in *NeMLaP3 / CoNLL98: New Methods in Language Processing and Computational Language Learning, ACL*, David M. W. Powers, Ed., January 1998, pp. 79–82.

General Scientific Premises of Measuring Complex Phenomena

H.M. Hubey, Professor
Computer Science
Montclair State University, Upper Montclair, NJ 07043

ABSTRACT

General scientific and logical premises lurking behind the art of measuring complex phenomena, specifically intelligence, are explored via fuzzy logic, probability theory, differential equations, thermodynamics, generalized dimensional analysis, philosophy and psychology.

KEYWORDS: *generalized dimensional analysis, path functions, fuzzy operators, fuzzy logic, thermodynamics, extensive variables, intensive variables*

0. THERMODYNAMICS, OSs, AND TURING

Thermodynamics is probably the classical and ideal example of a *system-theoretic* point of view, and one that is built on the twin concepts of *state* and *process*. Furthermore, it is probably the only link from physics to the study of living things, which are most likely the most complex things which humans will ever have to study. The physical sciences are the easy sciences; it is the life sciences that are the hard sciences.[1] Unfortunately, physical scientists work with powerful tools, and life sciences have restricted themselves to working with much less powerful tools[1].

Thermodynamics is a perfect example of a science whose development lead to the improvement of the measurement of a fundamental dimension of physics. It was not until Lord Kelvin saw some inconsistencies that the concept of an 'absolute' temperature scale was created. In measurements of things such as length, mass, or time we can easily envision the concept of 'zero'. But it is not so with temperature. Nobody knew what the lowest obtainable temperature was. In the arguments in the philosophy of science there exist *data-first* and *theory-first* schools. Here we have a case in which both are iteratively used. The problem of intelligence is most likely to follow this pattern of development. If the problem is in an area that has a well-developed theory, we must try to explain the phenomenon in terms of the developed theory. It is only when we cannot that we can start thinking about a new theory, and this requires datamining techniques.

An Operating System (OS) is a very complex object. It has been said that "I may not know what an OS is but I can recognize one, when I see one!". The same thing may be said about intelligence, (or cognitive ability or any of the other related words such as awareness, consciousness, or autonomy, or even life.) The Artificial life newsgroup (alife) skipped trying to define life or artificial life. The only serious effort in this direction was made by Alan Turing. He essentially formalized the saying about the OS into intelligence. We may not know what 'intelligence' is but we know how to recognize one when we see one. Apparently when we talk about intelligence, we are talking about 'human kind' or 'human type' or 'human level' intelligence, or at least 'living thing' kind (type/level) of intelligence. We can say things about this without being able to define it precisely. It is precisely about this intelligence that Turing was referring to when he wrote about what is now referred to as the 'Turing Test'. He understood all the problems that involve discussions of this thing called intelligence many decades ago and offered his 'Gordian Knot' solution. Sometimes thinkers are unable to break through the boundaries of what has been created. Whitehead claims that Aristotle hindered the development of science for 2,000 years because nobody was courageous enough to break through the boundaries of the box for the sum total of all knowledge for human kind.

1. MEASUREMENT THEORY I

Normally, in the physical sciences, the possibility that an instrument may be capable of high precision while not being able of high accuracy does not occur to people. It can only occur if the instrument is broken. If the instrument is a very simple one (such as a ruler) we'd see immediately if there was something seriously (or obviously) wrong. If the instrument is a highly complex one, then there would be various self-tests. However, in the social/life sciences creation of 'instruments' is an art. It is quite possible for the instrument to be *reliable* (precise) but not *valid* (not accurate) or vice versa. For example, a psychologist might decide to create a questionnaire which he claims measures 'hostility'. The same person taking this test (the questionnaire) might obtain different scores at different times. So habituated are we to measuring things in this modern age that we scarcely give thought to the possibility that what is being represented as a number may be meaningless. That is the validity of the measurement i.e. that the measurement or metric actually measures what we intend to measure. In physical measurements there is usually no such problem. Validity also comes in different flavors such as construct-validity, criterion-related validity, and content-validity. Reliability refers to the consistency of measurements taken using the same method on the same subject. *(Please see Figure 1)*

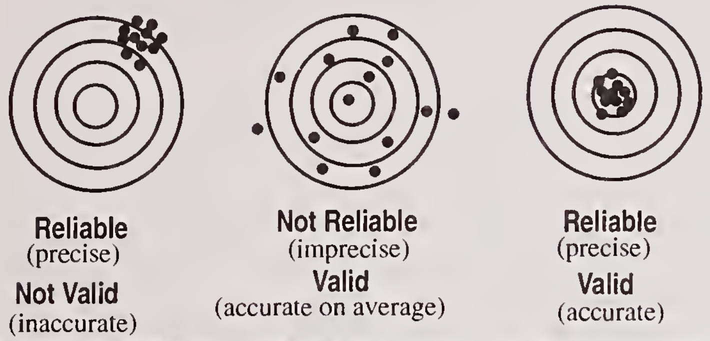


Figure 1: Reliability and Validity Analogy: One normally expects accuracy to increase with precision. However in the social sciences they are independent.

2. MEASUREMENT THEORY II

We often need to make things comparable to each other. We call this *normalization*. That is most easily done if we use numbers. For example, one way to normalize test grades is simply to divide every grade by the highest grade in class. This guarantees that the highest grade in class is 1.0 In order to be able to compare one boxing match to another a standard scoring system is used in which the same number of referees are used to score the bout, and for each round at least one boxer must be given 10 points. In Rasch measurements, we use

$$\frac{P}{1-P} = e^{\alpha - \delta} \tag{1}$$

where P=Prob{answering correctly}, α=ability, and δ=difficulty of question. However, this is not scale-free. It would probably be better to use something like

$$\frac{P}{1-P} = \frac{\alpha}{\delta} \quad \text{or} \quad \frac{P}{1-P} = 1 + \ln\left(\frac{\alpha}{\delta}\right) \tag{2}$$

In this case it is only necessary that both α and δ be measured on the same scale (somehow). Obviously, it would be best for all purposes to use numbers in the standard interval [0,1].

3. MEASUREMENT THEORY III

Before we try to normalize quantities we should know what kinds of measurements we have. They determine if we can multiply those numbers, add them, or can merely rank them etc. Accordingly measurements are classified as: (i) Ratio scale, (ii) Interval scale, (iii) Ordinal scale, or (iv) Nominal scale.

Absolute (Ratio) Scale: The highest level of measurement scale is that of ratio scale. A ratio scale requires an absolute or nonarbitrary zero, and on such a scale we can multiply (and divide) numbers knowing that the result is meaningful.

Interval Scale: The Fahrenheit and Celsius scales are interval scales. The differences on these scales are meaningful but ratios are not. That is what Kelvin found out, and that is what the absolute temperature scale is about. When measuring things such as intelligence, consciousness, awareness, or even autonomy, or hostility, we have no guarantee that we are measuring any of these on an absolute scale. There must be some other guidelines. One of the guidelines is obviously the study of various scales. In the intelligence game, psychologists have mainly relied on the *central limit theorem* in 'hoping' that intelligence is a result of many many different things adding up to create a Gaussian density. Thus they have contrived to make sure that test results are Gaussian.

Ordinal Scale: The next level on the measurement scale is the ordinal scale, a scale in which things can simply be ranked according to some numbers but the differences of these numbers are not valid. In the ordinal scale we can make judgements such as A>B. Therefore if A>B and B>C, then we can conclude that A>C. In the ordinal scale there is no information about the magnitude of the differences between elements. It is possible to obtain an ordinal scale from questionnaires. One of the most common, if not the most common is the multiple-choice test, called the *Likert scale*, which has the choices: extremely likely/agreeable, likely/agreeable, neutral, unlikely/disagreeable, and extremely/very unlikely/disagreeable.

Nominal Scale: The lowest level of measurement and the simplest in science is that of *classification* or *categorization*. In categorization we attempt to sort elements into categories with respect to a particular attribute. It ranks so low on the scale that it was added to the measurement scales later. Even an animal that can tell food from nonfood can be said to have learned or can be said to know about set operations instinctively.

The most basic and fundamental idea underlying these scales which is not even mentioned, and which is extremely important for measurement of complex phenomena in the life sceinces, is that in the final analysis, it is the human sensory organs that are the beginnings of all measurement. In the measurement of temperature, although a difference scale was easy to set up via the human sensory organs (and induction), it took theory and scientists to obtain an absolute scale for temperature. To obtain a difference scale the only thing necessary was for humans to note that the liquid in the glass went up when it was hotter. There was no way to know which was more hot and which less hot except via our naked senses.

This is/was as basic as knowing the difference between which of two sticks is longer than the other or which of two weights is the heavier one. Similarly in the measurement of intelligence, the final arbiter is still the naked human senses. Humans must make up the tests and decide which is more intelligent, say a chimpanzee or a dog. There can be no other way to proceed. The genius of Turing was that he realized this immediately. Therefore, Turing's basic intuition is correct. We might not know what intelligence is but we can recognize it when we see it. Secondly, we should probably turn to nature to find examples and a hierarchy or scaling of intelligences. It would not be off the mark to accept that all living things are intelligent to a degree, and that EI (Encephalization Index) is basically a good scale on which to compare the intelligences of at least some living organisms.[2]

4. MEASUREMENT THEORY IV

Before we can even think about whether our measurements are on an absolute or difference scale we have to make sure that the objects that we deal with are *quantifiable* in some way and that we can measure them (with numbers naturally). Our handle on the problem is that the things we measure in physics (and hence engineering) come in *fundamental dimensions*. For example, dimensions of that particular branch of physics called mechanics consists of M {mass}, L {length}, and T {time}. For electrical phenomena we need one more dimension, Q (charge), and for thermal phenomena we need θ (temperature).

Then we can entertain the thought of using *dimensionai analysis* for complex phenomena which is a method of reducing the number and complexity of experimental variables which affect a given physical phenomenon, using a sort of compacting technique. If a phenomenon depends upon n dimensional variables, dimensional analysis will reduce the problem to only k dimensionless variables, where the reduction $n-k=1,2,3$ or 4 depending on the problem. Since these new dimensions are products/ratios of the old variables to various powers, the new dimensionless space has nonlinearly twisted and compacted the old problem in a way in which we can see regularity.

These ideas have been put to good use in biology [3]. For example, the mass of an animal grows proportional to L^3 but its surface area is only proportional to L^2. Thus, as animals get larger they have to have larger cross-sections of bones to support all that weight. So an elephant does not look just like a large sheep. These ideas have to be taken into account when prototypes, say, airplanes are tested in wind tunnels. Many other things having to do with scaling of living things such as metabolism, oxygen consumption, heat exhaustion, cooling etc. can be found in Schmidt-Nielsen[3]. For example, one way to make different animals's brains comparable is to compare not their brain capacities but the ratio of their brain mass, b, to their body mass B. Until recently, there was no method that could cluster the variables in similar ways as above so that nonlinear dimensional compaction was not available, but now there is a generalized data-driven method.[4]

5. PHILOSOPHY

Why do we do philosophy? One reason is because we do not want to 're-invent the wheel'. If philosophers have already thought about this topic, we should at least be aware that thought has been expended and results have been achieved.

Operationalism: The problem of what is being measured in quantum mechanics was solved during the early part of this century by 'operationalism' an idea (by Bridgeman) that the operations that are being executed define what is being measured. As long as everyone does the same thing, we are guaranteed that we all measure the same thing. In the measurement of something like intelligence, obviously, the problem of validity remains.

Quality vs Quantity: Thermodynamics, gave us the concept of *extensive* and *intensive* variables. It is often remarked in narratives that a fundamental difference exists which can be characterized by the words 'quantitative' vs. 'qualitative'. Often what is meant by the word qualitative is "intensive" since concepts often characterized as a quality can also be quantified. If a system consisting of a lot of 10,000 TVs is split into two sets at random, the quality of the two subsystems will equal each other and the quality of the TVs of the whole original system. A state of a system is characterized by a set of parameters. If we split a thermodynamic system (say a container of gas) in half some of the parameters will obey $X_1 + X_2 = X_s$ and others will obey $x_1 = x_2 = x_s$. The former (upper case) are *extensive* parameters, and the latter *intensive* parameters.

Open vs Closed: The concepts open vs closed (*endogeneous vs exogeneous*) are obviously very closely related to each other. In a closed system there can be no such thing as an exogeneous variable. At the same time, in general there is really no accurate or clear definition of what an open system is. In thermodynamics from where these ideas are probably borrowed, an open system is one which exchanges mass with its surroundings. A closed system may exchange heat, and do work on its surroundings, or have work done on it by its surroundings. Additionally, heat and work are processes. In other words, they are not *point functions*, but *path functions*.

In general in mathematical modeling via differential equations, the surroundings (*forcing* or *source* term) is everything that does not have the system variable in it and usually put on the rhs. However, when these concepts are specifically applied to intelligence, we have to clarify what it is that the system exchanges with its surroundings. The concept can apply to both exchanging data and or information with its surroundings. At the same time, the word "open" may be used to refer only to the problem at hand (i.e. if the problem is "open-ended"), but then it is not about generalized intelligence but about a specific problem. To generalize it we will then be forced to think about what little we know about how the brain does its work or how to generalize from the mathematical methodology that presently exists (i.e. logic, probability theory, etc). [1]

Many-as-One: The most fundamental such concept according to modern math is 'set' and forms the basis of logic, where philosophers are at home. This idea is the building block of all systems. A body is not just a parts list although it is comprised of many subsystems thus is not merely a set. We have many ways in mathematics of treating many things as one. A *tensor* is a general object of any degree. A zero dimensional tensor is a *scalar.* A one dimensional tensor is a *vector* or an *n-tuple.* A two dimensional tensor is called a *matrix.* In addition to this, from computer science we have the latest, and more flexible concept of hierarchical ordering via OOP (object-oriented programming) in which an object is a set of parameters without necessarily being merely a set or a vector.

Parallel vs Serial (sequential): This is one idea that occurs quite often. Some problems are parallelizable. For example, to dig a large ditch if we hire 100 workers as long as they do not interfere with each other, the ditch-digging will go at a rate 100 times as fast as before. However, if I want to send a message with a messenger, it does not matter if we use 100 messengers. The increase in the number of messengers might increase the *reliability* but will not affect the speed of the delivery. But parallelity also has to do with simultaneity (not always in time), choices, and substitutability, and logic.[7]

Trade-offs and Logic: We can sometimes trade-off something for something else in which case these things are substitutes of some kind. This idea shows up in logic as a logical-OR (co-norm). In the psychology and cognitive science literature, many different components of intelligence are posited. It is quite possible that some of these intelligences are composed of other more primitive types. If so, then are some of these substitutes for each other?

6. PSYCHOLOGY & COGNITIVE SCIENCE

Obviously throughout most of the century those who have worked on the nature and measurement of intelligence (almost always human intelligence) have been psychologists. They have had recourse to and benefited from methods and argumentation in both philosophy and physics. The kinds of questions with which they have toiled can be summarized in modern (and mathematical) terms as:

i) What kind of a quantity is intelligence? Is it *binary* or measurable on some scale? What kind of a scale is appropriate? Is it an *ordinal*, *interval*, or an *absolute* (ratio) scale?

ii) Is it an *additive function* of its constituents, the most important ones for purposes of simplification being hereditary (nature) and environmental (nurture)? Or is it a *multiplicative function*? Is it logarithmic function, an exponential function or a polynomial function of its variables?

iii) Is it a *vector/tensor* or a *scalar* (Spearman's g)? In other words, can a single number be produced from many numbers which is meaningful? Is there a hierarchy of intelligences, some of which subsume some of the others?

iv) Is it a *state* or a *process* ? In other words is it a *point function*, or a *path function*? Is it a *quality* or a *quantity*? In other words, is it an *extensive* variable or an *intensive* variable?

v) The *nature vs nurture* problem: Are the differences in intelligence among humans due mostly to heredity or environment?

There is a related (and incorrectly stated) version of (v) which is "Is intelligence mostly genetic?" The answer is quite plainly that intelligence is mostly genetic if intelligence is discussed in its most general form, that is including machine intelligence and animal intelligence. However the answer to (v) is much more complicated.[5]

An almost perfect example of a vector of cognitive science is color. We all know what colors are but they would be virtually impossible to explain to someone who was congenitally blind. If we did attempt to "explain" colors by explaining that "black is the absence of color and white is a mixture of all the colors" it is likely that the blind person would think of colors as what we call "gray scale". The analogical question is whether the components of intelligence that psychologists have posited are like colors in that they 'seem' as if they are 'unique' objects or is there a single number which we may obtain from the components.[8] Is this single number like colors or is it like the gray-scale?

7. COMPLEXITY AND HIERARCHY

The concept of layering or hierarchy is one of the most basic in the universe. Whereas hierarchy requires more detailed explanation the concept of layering is easier to envision and observed all over the world, at a very coarse-resolution. We use pictures of all sorts (as in Figure 2).

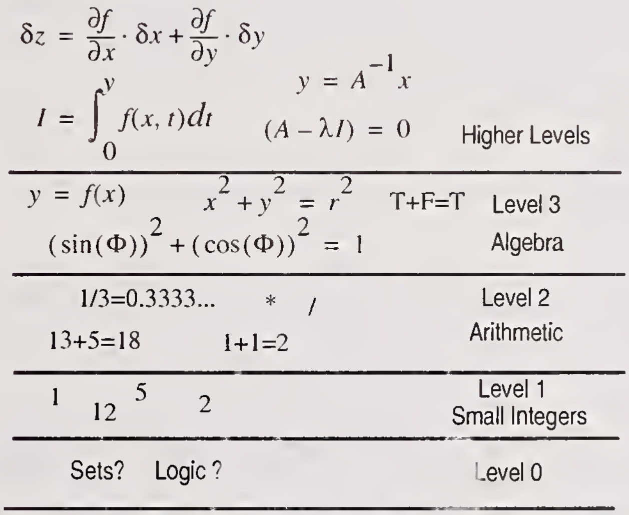


Figure 2: Highly-suggestive Layering in Mathematics: Knowledge is built-up in layers. New knowledge is built on top of old knowledge. This has significance for intelligence testing.

What better example than knowledge? *Data is raw. Information is data that is meaningful to an intelligent entity. Knowledge must be compressed information.* The only way to compress information is via exploiting regularities and patterns. Since mathematics is the study patterns, and regularities of all kinds, it is clearly the best tool with which to do science. Many more examples of layering can be found [1],[5],[6].

Thus the *scientificity* (intensity) of knowledge must be mathematics. Is it possible to measure intelligence separate and apart from knowledge? Do we want to weight some kinds of knowledge more heavily than others?

8. DISTANCE & MEASUREMENT

The main problem here is whether, after having gone through the problem of identifying the various components of intelligence, we should multiply them or add them to create a single number called intelligence. Therefore two prototypical choices for distance are

$$d(\vec{x}, \vec{y}) = \left(\sum_{i=1}^{n} (\alpha_i x_i - \beta_i y_i)^{2m} \right)^{\frac{1}{2m}} \qquad (3)$$

$$d(\vec{x}, \vec{y}) = \left[\prod_{i=1}^{n} x_i^{\alpha_i} \bullet y_i^{\beta_i} \right]^{\frac{1}{\Omega}} \qquad (4)$$

Obviously, in Eq (4) every component must be nonzero. There are good reasons why it is so. If normal functioning of a human depends on having absolutely no genetic defects, and if the intelligence of a human is determined by n genes, then if any of them is defective it should effect the score in the same way that the reliability of a composite is the product of the reliabilities of its components. In this sense, then the factors are analogous to probabilities.

This is also how we humans apparently tend to evaluate intelligence, as can be seen in the schizoid labeling of the condition known as *idiot-savant.* Being apparently superhuman in one aspect of intellectual activity is not sufficient to escape the label 'idiot'. It is said that *an expert knows everything about nothing whereas a generalist knows nothing about everything.* In an extension of this, then, today's experts (i.e. engineers) are idiot-savants. Their social IQ is said to be low. Programs like Maple, then, are also idiot-savants.

9. AVERAGE-IZATION

Consider the problem of being a juror in a beauty pageant. We will be forced to use a kind of scale in Eq. (5) (below)

$$B(\vec{x}) = 1 - \left(\prod_{j=1}^{n} \left[\{x_j - \mu_j\}^{\alpha_j} \right] \right)^{\frac{1}{\Omega}} \qquad (5)$$

where the μ_i are the means. For example, the features/properties (of the vector x) may be nose length, skin color, lip thickness, fatness, etc. We will not want to vote for those with lips too thin or too thick, with noses that are too long, or too short, legs too thin or too thick, skin too pale or too dark. In other words, we are not looking for the minimum or the maximum but rather the most perfect average there is (with some caveats). This is a different kind of logic, triage logic [10].

Then, the human-kind of intelligence, if it is going to resemble what we humans normally think about perfection (apparently) should be measured via

$$I(\vec{x}) = 1 - \left[\prod_{i=1}^{n} \{x_i - \mu_i\}^{\alpha_i} \right]^{\frac{1}{\Omega}} \qquad (6)$$

where the {x} are the various attributes of intelligence. The Turing test is probably for this kind of intelligence. For example, a machine that can solve differential equations and multiply 20 by 20 matrices in a jiffy (such as Maple, a Computer Algebra System) would flunk the Turing test. A human would know that a normal human (or maybe even an abnormal human) cannot do that. Therefore, the machine that could pass the Turing test would either have to be designed dumbed-down or it would have to learn to deceive. There are other things machines can do very quickly that humans cannot accomplish.

Thus the 'measure' above would show that such an entity could not be human (ceteris paribus, of course). In other words, as long as the machine is able to do the other things more or less as a human, then overachieving (outdoing humans) in one of the dimensions of the vector space would mark it as a machine.

Exactly the same would apply in some other capability such as being able to lift a few tons, swimming or running at superhuman speeds etc. For machines, then locomotion, would also be treated as part of intelligence. However, since even lower animals (less intelligent than us) can move around, it should not contribute much to the measurement of intelligence.

There are some pyschologists who want to include many human capabilities, such as physico-kinetic intelligence (i.e. physical ability) in the intelligence equation. Therefore, this 'autonomy' capability of animals/machines may also be considered to be a part of intelligence. We may take those that have been posited by psychologists as a starting point keeping in mind that some of them may really be substitutes for each other so that the measurement might be more complicated.

10. MORE SOPHISTICATION

Consider the simple problem of nutrition. Suppose we can create a balanced diet from the few foods available from three separate food groups; meat (protein), carbohydrates, and vegetables as shown below.

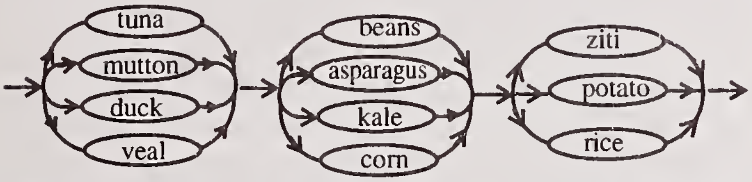


Figure 3: Parallel or Serial Choices. The problem is actually about multiplication vs addition. Diagrams such as this occur in electrical circuits, Boolean circuits [9], or choice making.

In terms of circuit analysis (which can be thought of in terms of Boolean algebra, [9] it is clear that the parallel lines are about choices (and thus lack of constraints) and therefore represent logical-OR (disjunction), whereas the seriality/sequentiality denotes a logical-AND (conjunction). Probably the first thing a statistician would do if faced with the problem of determining the relationship between food groups and a balanced diet would be to try correlation-regression analysis which would be nothing more than

$$N = \alpha_0 + \alpha_1 t + \alpha_2 m + \ldots + \alpha_n c \qquad (7)$$

where t=tuna, m=mutton, c=corn etc. This is really the same kind of valuation of the problem as a weighted average. However, if we think logically then we should be considering a function of form;

$$N = MCV \qquad (8)$$

since we need to ingest food from all the groups. Furthermore, since these food groups may be instantiated via specific examples, then using fuzzy logic, we should be regressing one of

$$N = (t+m+d+v)(b+a+k+c)(z+p+r) \qquad (9a)$$

$$N = (t+m+d+v)^{\alpha}(b+a+k+c)^{\beta}(z+p+r)^{\gamma} \qquad (9b)$$

Obviously, the latter form (Eq. 9) is not only correct but will result in many products (possibly to various powers). It is exactly this kind of products that dimensional analysis produces however it works only for problems with physical dimensions. However, there are methods that will produce similar equations for any problem if sufficient amount of data is available [4]. If intelligence-measurement is at least as complex as that of proper nutrition, then the simple weighted average kind of methods which are additive will not work. In other words the regression in Eq (7) is something like a combination of logical (or fuzzy) ORs and ANDs. A question that comes to mind is if there are fuzzy operators which are neither OR nor AND but something like both and exactly like neither. The special functions [11]

$$H_h(x, y) = \frac{1}{2} \cdot (x+y)^{h+1} \qquad (10a)$$

$$M_m(x, y) = 2^m \cdot \left(\frac{(x-y)^2}{2[(x-y)^2]^{1/2}} \right)^{m+1} \qquad (10b)$$

or others similar to these can be used in cases in which we are not sure if additive or multiplicative models should be used. One can show that [11]

$$Max(x, y) = H_o(x, y) + M_o(x, y) \qquad (11a)$$

$$Min(x, y) = H_o(x, y) - M_o(x, y) \qquad (11b)$$

Therefore the operator (fuzzy *t-co-norm*)

$$F(x, y) = H_o(x, y) + (2\xi - 1) M_o(x, y) \qquad (11c)$$

is neither a norm (intersection) or conorm (union) but a fuzzy operator or a fuzzy norm since it is a norm for $\xi = 0$ and a conorm for $\xi = 1$. Some of the present day attributes of intelligence posited by psychologists probably are substitutes for each other and thus Eq (6) might distort the measurement. Therefore, something like Eq (9) where the additions are fuzzy unions and fuzzy intersections will probably give better results. The equations are readily and intuitively comprehensible in terms of theory of reliability based on probability. Fuzzification of the norm-conorm can be done for any fuzzy logic. For example, the simple product/sum logic given by

$$i(x, y) = xy \qquad (12a)$$

$$u(x, y) = x + y - xy \qquad (12b)$$

can be easily fuzzified via

$$F(x, y) = \rho xy + (1-\rho)(x+y-xy) \qquad (12c)$$

11. HUMAN INTELLIGENCE

The main problem today in human intelligence tests (and genetics) is calculating how much of intelligence is ‘inherited’ and how much of it is learned. There are several ways in which the model for this may be derived. One way would be to point out general conditions which the ‘intelligence function’ must satisfy. It should be multiplicative. It should display the increase of intelligence in time from the time of birth. It should converge on some limit on average for the people while being allowed to fluctuate about the average rate of increase and the limit of human intelligence. The equation

$$\frac{dx}{dt} = \lambda(\alpha - x) \qquad (13)$$

increases exponentially, and converges to a limit which is a good approximation. We need to know what the parameters mean, and this can be gleaned from the behavior of the solution. In Fig (4a) we see several trajectories. Some converge to above average intelligence, and some to less than average. Obviously the coefficient α determines this limit.

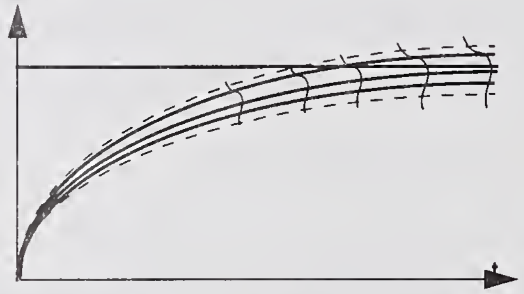


Figure 4a : Variations in α of the Intelligence Model .

In Fig (4b) we see a fluctuation in the rate of increase of intelligence, and this is controlled by the coefficient λ .

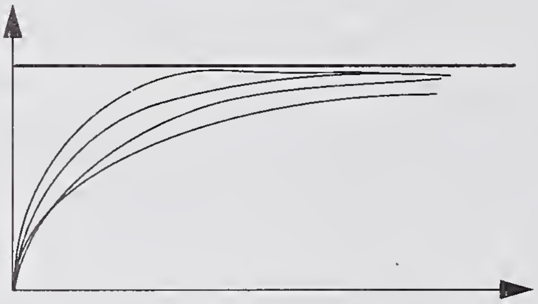


Figure 4b: Fluctuations in λ of the Intelligence Model

Logically both of these parameters then should be a function of both genetics and environment. Since we have determined that multiplicativity is important, the model should be

$$\frac{d}{dt}I(t) + \lambda G^{\eta}E^{\varepsilon}(t)I(t) = \lambda\alpha G^{h+\eta}E^{e+\varepsilon}(t) \qquad (14)$$

Integrating it once and rearranging terms we obtain the integral equation

$$I(t) = K(t) - \lambda G^{\eta}\int_0^t E^{\varepsilon}(\sigma)I(\sigma)d\sigma \qquad (15a)$$

$$\text{with } K(t) = \alpha\lambda G^{h+\eta}\int_0^t E^{e+\varepsilon}(s)ds \qquad (15b)$$

which is exactly what most researchers claim, that is, intelligence at time t, that is I(t), is a function of the past interaction of intelligence with environment summed up over time from time zero (birth) to the present time t. The interaction is multiplicative as it should be, and the equation is a reasonably good approximation over time of how living things (especially humans) learn. The solution is

$$I(t) = \Gamma e^{\lambda G^{\eta}\int_0^t E^{\varepsilon}(\tau)d\tau}\int_0^t E^{e+\varepsilon}(s)e^{\lambda G^{\eta}\int_s^0 E^{\varepsilon}(\tau)d\tau}ds \qquad (16)$$

where $\Gamma = \lambda\alpha G^{h+\eta}$ which in the limit goes to

$$I = \alpha E^e G^h \qquad (17)$$

If one day robots which learn from their environment are created, similar equations will be good first order approximations. Same probability techniques can be used on these equations, and statistics such as 'heritability' can be calculated. If the multiplication above is treated as some kind of a fuzzy intersection, then we can see quite clearly that the same kind of an equation can easily 'explain' the existence of natural language among living things. At the limits the equation must reduce the crisp logic, and we can see that it does. Only in the case when both genetic capability is there and when there is proper environmental stimulation, does language exist. If one or the other is missing there is no language. We can show how this equation explains what psychologists have said (in words) for a long time. Computing the virtual variation, we obtain for the special (and simpler case) of $e = h = \alpha = 1$

$$dI = E \cdot dH + H \cdot dE \qquad (18)$$

If the environment is enriched, the corresponding increase in intelligence depends on the genetic capability. Thus putting a dog in school cannot give it human level intelligence. Similarly, if there is a change in the genetic make-up (e.g. the difference between a chimp and a human) the change in the intelligence depends on the environment. A human brought up without human contact cannot walk or talk or dress up.

APPENDIX

Exact Differentials and Path Functions

The distinction between the related concepts *state* and *process* is an important one. There are mathematical definitions and consequences of these ideas. A *state* (or property) is a *point function*. The state of any system is the values of its *state vector* (a bundle of properties which characterizes a system). If we use these variables as coordinates then any state of the system is a point in this n-dimensional space of properties/characteristics. Conversely each state of the system can be represented by a single point on the diagram (of this space). For example for an ideal gas the state variables are temperature, pressure, volume, etc. Each color can be represented as a point in the 3-D space spanned by the R, G and B vectors. Intelligence is commonly accepted to be a state variable, i.e. a point. The scalar, Spearman's g, (single number, not a vector) can be obtained from this vector by using a distance metric. The argument that the values of the components cannot be obtained from the scalar, g, may be valid depending on the distance metric however, the distance metric may be devised in a way in which the components can be obtained from the scalar. Distance on a metric space is a function only of the end points i.e. between two states. However, the determination of some quantities requires more than the knowledge simply of the end states but requires a specification of a particular path between these points. These are called *path functions*. The commonest example of a path function is the length of a curve. Another example is the work done by an expanding gas. So is Q, the heat (transferred). In that sense work and heat are interactions between systems (i.e. processes), not characteristics of systems (i.e. state parameters/variables). Intuitively, when we talk about small changes or small quantities we use the differentials dx or δx. However the crucial difference is that although there may exist a function such that

$$\int_b^a dF = \int_b^a f(x)dx = F(x)\Big|_b^a = F(a) - F(b) \qquad \text{(A.1)}$$

there is no function Q, (heat) such that

$$\int_b^a \delta q = Q(x)\Big|_b^a = Q(a) - Q(b) \qquad \text{(A2)}$$

Instead we write

$$\int_a^b \delta q = Q_{ab} \qquad \text{(A3)}$$

meaning that Q_{ab} is the quantity of heat transferred during the process from point *a* to point *b*. Similarly because the infinitesimal length of a curve in the plane is given by

$$ds = \sqrt{dy^2 + dx^2} \qquad \text{(A4)}$$

we cannot integrate ds to obtain

$$S(b) - S(a) = \int_a^b \delta s = \int_a^b ds \qquad \text{(A5)}$$

but instead first the curve y=f(x) must be specified. Equivalently, if z is a function of two independent variables x and y, and this relationship is given by z=f(x,y) then z is a point function. The differential dz of a point function is an *exact differential* and given by

$$dz = \left(\frac{\partial z}{\partial x}\right)dx + \left(\frac{\partial z}{\partial y}\right)dy \qquad \text{(A6)}$$

Consequently if a differential of form $dz = Mdx + Ndy$ is given, it is an exact differential only if

$$\frac{\partial M}{\partial y} = \frac{\partial N}{\partial z} \qquad \text{(A7)}$$

Therefore in the mathematical function used for the simple two-factor (nature-nurture) *Intelligence Function* the *environmental path taken* does make a difference in the final result which is assumed to be a state function (although computed from mental processes).

REFERENCES

1. Hubey, H.M. (1996) Topology of Thought, CC-AI: The Journal for the Integrated Study of Artificial Intelligence, Cognitive Science, and Applied Epistemology, vol 13, No.2-3, 225-292.

2. Eccles, J. (1989) Evolution of the Brain: Creation of the Self, Routledge, New York

3. Schmidt-Nielsen, K. (1984) Scaling: Why is Animal Size So Important?, University of Cambridge Press, Cambridge.

4. Hubey, H.M. (2000) A Complete Unified Method for Taming the Curse of Dimensionality in Datamining and Allowing Logical-ANDs in ANNs, submitted to Datamining and Knowledge Discovery.

5. Hubey, H.M. (1994) Mathematical and Computational Linguistics, Mir Domu Tvoemu, Moscow, Russia.

6. NIST (2000) White Paper

7. Hubey, H.M. (2001) Evolution of Intelligence:Direct Modeling of Temporal Effects of Environment on a Global Absolute Scale vs Statistics, accepted by Kybernetes: The International Journal of Systems and Cybernetics.

8. Hubey, H.M. (1996) Logic, physics, physiology, and topology of color, Behavioral and Brain Sciences, vol 20:2, pp.191-194.

9. Hubey, H.M. (1999) Mathematical Foundations of Linguistics, Lincom Europa, Muenchen, Germany.

10. Hubey, H.M. (2000b) Fuzzy Logic and Calculus of Beauty, Moderation, and Triage, The Proceedings of the 2000 International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences (METMBS2000), June 26-29, Las Vegas.

11. Hubey, H.M. (1999) The Diagonal Infinity, World Scientific, Singapore.

Evaluation of System Intelligence via Pictorial Data Visualization

V. Grishin*, A. Meystel°
* View Trends, Ltd., Cleveland, OH
° Drexel University, Philadelphia, PA

Extended Abstract

1. Introduction: Aspects of the research

The concept of evaluating the intelligence of systems presented in this paper is based upon the model of intelligence outlined in [1] and the advancements in visualization described in [2]. Since the main mechanism of intelligence is the mechanism of generalization, it would be prudent to judge the degree of intelligence by the ability of the system to generalize. This ability can be detected by the means of visualization. Visualization of the system and/or the situation allows us to use the primary orientation of our visual capabilities to the situations and/or modes of functioning based upon "gestalt" i.e. capabilities to form a harmonious and consistent entity out of details.

We will explore the unique ability of the visualization systems to diagnose the system and/or its state by discovering the *syndrome*: a group of symptoms, or diagnostic features that collectively indicate or characterize a disease, a disorder, or another abnormal condition which has some unity within itself. We will use the term syndrome for technological cases either to characterize some psychological situation based upon an intrinsic or other unity. For example, a multiplicity of unfortunately coinciding factors can lead to a catastrophe. Thus, for this particular catastrophe, the combination of these factors is a *syndrome*.

Thus, our approach is pursuing two major goals. First, our intention is to solve the unsolved yet a fairly complicated problem of data mining and interpretation. This is a central problem of intelligence functioning: *how knowledge can be extracted from raw data* via visualization. Solving this problem would require analysis of the real world situations and constructing their models by effectively combining formal, verbal and even non-verbalized models of analyzed knowledge. It turns out that visualization can help the human decision maker to associate these diversified models and to formalize the new knowledge for the subsequent use in both manned and unmanned intelligent systems.

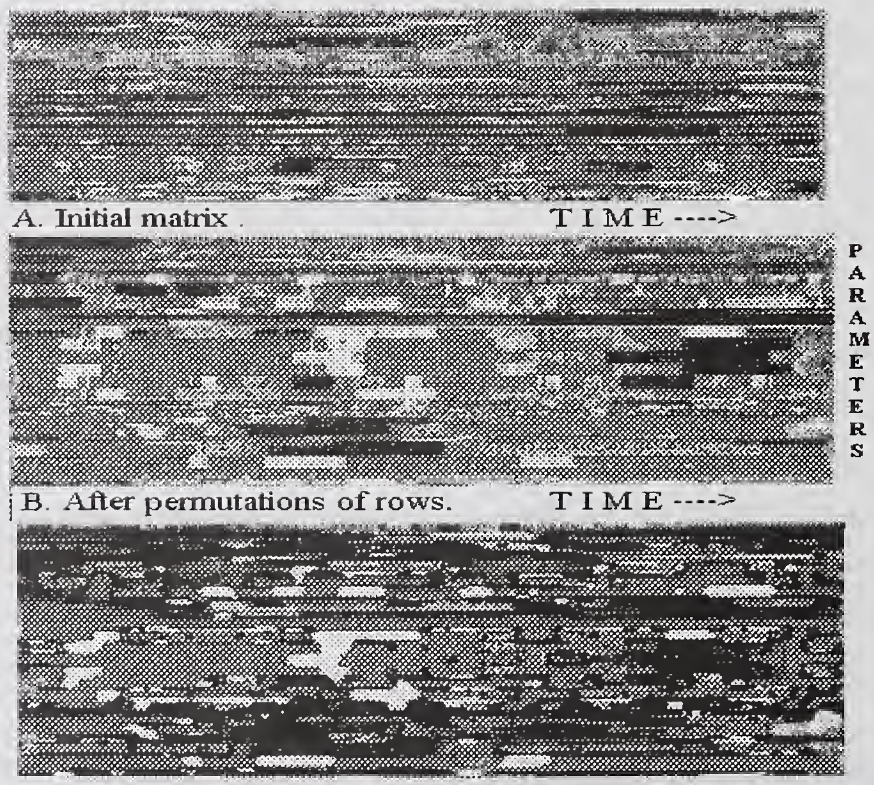
To accomplish this goal a human-computer dialog has to be constructed at each step of visualization. This dialog is aimed into restoration and analysis of the *hierarchy of features and descriptions* for states, situations, and scenes. It should provide for fast and accurate discrimination, description and understanding of *known and new* situation and their reasons. A visual-verbal language is created as a

part of this dialog for each case of analysis individually. This approach provides effective selection of new models for discovered singularities, observed changes of situation, detected structures, etc.
An effective *interpretation* of visual-verbal results in terms of data properties and known models is the result of this approach. Although the human participant cannot be replaced at this point: there is no automated procedures to rely upon. A number of important advantages can be registered in comparison with automated systems neural networks, pattern recognition, etc. To get the good interpretation we use the *simple data mapping into pictures* and the human natural "gestalt-skills" for determining entities in these pictures.

2. Principles of the Human-Computer Dialog for Picture Analysis

The following main components of our *dialog realization* distinguish our approach from others by more effective use of human cognition:

1. *Constructing holistic images of exhaustively represented data about situation.* If the number of variables in a situation is more than 20-30, the matrix N*M is to be analyzed where each row displays a time series of one variables or each cell represents a current value of separate variables or others. A variable value is mapped into color-brightness. The ability to simultaneously represent more than 1000*1000 numbers and to see some "general image" of situation is considered to be an advantage of our realization.



A. Initial matrix .

B. After permutations of rows.

C. After picture smoothing and sharpening of edges.

Fig. 1 Color matrix display of dynamics of 114 parameters of nuclear power unit for a small leak of steamgenerator. (220 time samples)

This starting image allows for understanding a holistic structure of the situation, for detection some of the available skeletons, and for mapping its different properties into levels of some hierarchical organization that will direct the subsequent informative feature search (see Fig 1 A).

2. *Combinatorial Searching in the Matrix* by permutations of rows, columns and cells. For each permutation the following is performed: sharpening the edges where required, smoothing where possible, value-to-color mapping adjustment where beneficial, etc. to get more informative, more interpretable image or at least to improve of its quality (Fig. 1 B,C).

3. *Mapping from matrix to entities: individual patches, group patches, clusters of groups.* Informative variables and features found as a result of matrices permutation can be visualized by grouping the elementary units of image together. Frequently, change of the coordinate system (e.g. from Cartesian to polar) can lead to new useful avenues in interpretation (as shown in Figure 2).

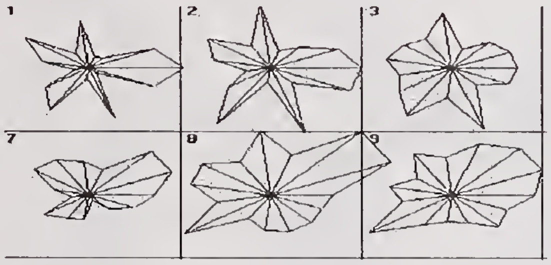


Fig. 2. Polar contour representations of 15 variables – "stars".

Thus, only tens of variables will be displayed at the levels of lower resolution but with complete mapping of their relationships within the image. This allows to determine shapes and more general forms that are more effective for visual analysis than color variation in the primary high resolution crowd of elementary patches. The dialog with generalized levels consists of a searching group variables and relationships among them.

4. *Selecting appropriate criteria of decisions.* It is important to underline that if no a priori knowledge and/or hypotheses exist, then forming a syndrome is done based upon human gestalt skills and experiences. If there is some knowledge of situation and its evaluation criteria, the process of interpretation will synthesize this knowledge with the gestalt intuitions.

5. *Synthesis of artificial representation of object* – has to show selected sindroms in a manner allowing effective application of above criteria.

3. Pattern Discovery by Human Vision: Can It Be Automated?

Human vision has an ability to quickly and efficiently compare several images in parallel with hundreds of local attributes and features related to the shapes, textures, colors, or brightness of the images. A standalone spot on a picture or a spot cluster has certain boundaries shapes which can be segmented using simple local features such as "straight line", "concave", "convex", "angle", "hole", etc. These features have attributes as "sizes", "orientation", "symmetries", etc. and are connected by means "upper-lower", "left-right", "inside-outside" adjacency. Local features are visually unified into more complicated shapes as "wave", "leaf", "a face profile" and others associated with real world objects and also have above mentioned and more complicated attributes.

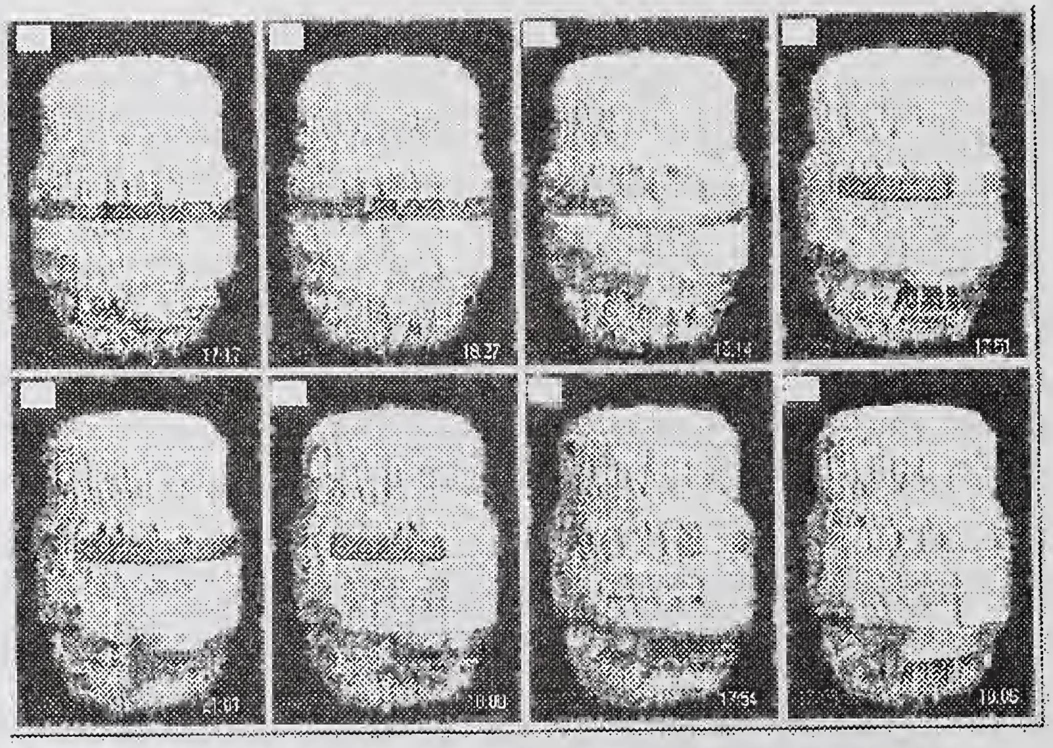


Fig. 3 **State changes of nuclear power during 62 hours are clearly visible on this artificial pictorial representation but they were not detected by standard control system of power plant. (dates shifted).**

This process of feature generalization continues up to holistic image of a spot including also its integral features as "complexity", "symmetries", "elongations" besides usual sizes, orientations, and position on the picture and others. In addition, the color and textural properties can be described. These attributes and features are then organized into a multilevel (multiresolutional) hierarchy that can be partially verbalized, or at least, tagged with symbols. If a picture contains many different separate shapes, such hierarchy can be constructed for these shapes clusters and clusters groups up to all picture. In addition, the combinatorial and statistical features could be visually detected and estimated. Vision rapidly moves through this hierarchy, searching for more details or generalizing the attributes allows for the simultaneous examination of many facets of the image by means of a variety of attributes and features.

4. Automated Description of Visual Patterns

Combinations of disjunctions and conjunctions of features Qi and their attributes Ai can be applied for formalizing human representation of patterns. So called conjunctive normal form (CNF) describes some pattern with variation of features attributes, e.g. [“middle size”[Qa **AND** Qb [which is “symmetrical” around axis Ac] **AND** Qk [known to be “small concave” and located in Ai (k) (e.g. position)]. Disjunctive normal forms (DNF = CNF_1 **OR** (Q_g v A_k) **OR** CNF_N) from separate features or complicated patterns describe picture classes with supplemental patterns. These descriptions are invariant for global rotations, shifts and projective transformations of whole shapes as well as their parts (with some limits). Similar formal tools can be applied with the purpose to formalize many other elements of the human-computer dialog. The transfer of knowledge from a human to a computer can be performed by using a subsystem of learning.

The analysis suggests that automated visualization is efficient in discovering entities, syndromes, and singularities. The following phases of the analysis can be focused upon.

Phase 1. Development of Automated Visualization System for Decision Making.
Usually data visualization is a *human-computer dialog* with the following general structure :
Stage 1. Entering data into the system and their consecutive processing in subsystems 1-4

Subsystem 1. Data gathering, transformation, filtration.

Subsystem 2. Mapping Data into visual paradigm, e.g. pictures.

Subsystem 3. Computer supported human visual analysis of the visualized data: features selection and transformation of situations into a visual relational map.

Subsystem 4. Comparison with a priori knowledge related to the features and the multi-feature formations and search of new ones.

Stage 2. Change of the chosen set of variables and parameters for the analysis and repetition of the cycle of consecutive running of Subsystems 1-4.
Stage 3. Estimation of results, hypothesizing entities (syndromes), testing it through Subsystems 1-4 again, formalization and decision-making.

These three stages are run presently as a human-computer dialog that can have cycles between these stages in any order assigned by a human. We intend to automate this process by equipping the human-computer dialog processes by learning subsystem.

The strategy and the techniques of implementing the subsystem of learning and subsequent conducting the interpretation of results will be determined by the following factors:

- *goals are pursued within a particular domain and assignment*
- *limitations of combining human and computer capabilities*

- *available algorithms of generalization and instantiation*
- *metrics accepted for evaluating the performance and intelligence of the system.*

Phase 2. Application of Automated Visualization System for evaluating performance and Intelligence of Intelligent Systems.

In this case, the result of learning from the human during the human-computer dialog will be used for both: a) automatic analysis of data and b) for evaluating the performance and intelligence of intelligent systems.

Assume, an intelligent computer vision system has performed image processing. As a result of this, a particular image underwent a multiple generalization and the results of this are presented as the result of image analysis and interpretation. Let us consider another case: an intelligent system has planned a motion trajectory for an unmanned vehicle. In order to evaluate the intelligence of these systems, their problem solutions are presented to the automated system of visualization. The structure of the image and the structure of the motion trajectory are visualized and the prospective syndromes are obtained. The results of visualization are compared with the results of processing by the system undergoing testing. This comparison serves as the estimate of performance and intelligence.

5. Visualization can be used not only for states but for the state-space trajectories

It seems natural to expand the process of visualization from evaluation of states and situations to evaluation of state space trajectories as a whole. This would allow for comparison of different system behaviors by means of visualization of appropriate data. The results of visualization in this case are not the *images*, or *pictures* but rather *movies*. There is plenty of evidence that the gestalt abilities can be applied not only to static images but also to their consecutive strings that represent *processes*. Finding a temporal unity of a process is the problem that has never be proposed before as a problem for the system of automatic visualization.

Intelligence is defined as a faculty of a system that increases the probability of successful functioning in a variety of problem solving situations and under uncertainty of the conditions of the environment.

When systems function, the results of their functioning reflect not only changes of the environments and the goals assigned but also the results of their control system generating decisions and shaping processes. The consistency of control system functioning will be reflected in a temporal gestalt of processes that are generated as a result of control. It is our hypothesis that one can judge the control system by observing the output and not only measuring how close it is to the output specifications but also how *satisfactorily* the system responds to all changes. Since, the construction of a metric that evaluates responses to all changes is a problematic one (H-infinity is one of the efforts) and since the combination of uncertain circumstances has unlimited number of possible combinations, we assume that using the natural ability of human vision to register and recognize *singularities* of external images, the ability to distinguish differences in response can be detected via visualization.

6. Existing Experience of Using Visualization for the Purposes of Recognizing Singularities

To the extent we analyzed the results of our experiences in Functioning Systems our hypotheses can be considered confirmed. Our experiences in visualization system development for human decision making support has shown that appropriate *data visualization* can :

- drastically enhance efficiency in comparing different approaches of intelligence,
- specify the most effective field of each approach application and combine many of them to built an intelligent system for wide diversity of environment variations and control tasks (or whatsoever…),
- extend this system capabilities for some set of important but uncertain (unpredictable) situations by means their holistic visualization and recognition in real time.

Gas-turbine engine diagnostics in airplanes, nuclear power unit monitoring and search for the cardiology diagnostic syndrome demonstrate capabilities of visualization techniques (see Figures 3 and 4 that illustrate the capabilities that arise during the analysis). Analysis of existing experimental data has allowed to expect that the proposed method of intelligence evaluation can be successful. Pictorial visualization has allowed to analyze the *transition* modes of equipments and temporal processes of human heart functioning. As a result, the effect of much earlier symptoms of many malfunctions in the *transition* modes of operation were discovered to be different from the static modes, and more reliability of conclusions was achieved.

In our interpretation of the cases of successful use of visualization, the following subjects were taken in account:

-What was special in the way we have arranged the process of visualization

-What does it imply for the future organization of visualization

-What are the “Hypotheses of Visualization” that can be formulated

-What are the new concepts that should be introduced: temporal gestalt, dynamic syndrome, visualization of transition modes.

The recommended use of visualization for intelligence testing include:

-The specifics of intelligence testing

-The similarities of the case of intelligence testing and examples

-Restatement of the Hypothesis of visualization for the case of intelligence evaluation.

- How it will be applied for the cases of

- planner/controller for industrial crane
- autonomous unmanned vehicle

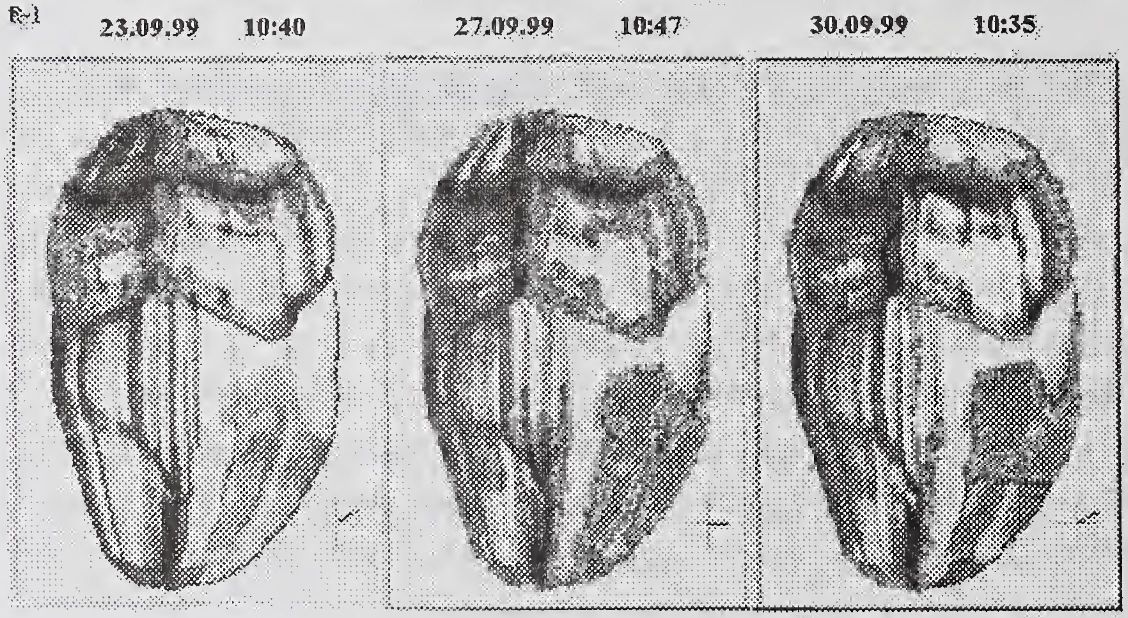


Fig. 4 Visible methabolic shifts - hypokallemia.
Features and dynamics of left atrium and ventricle enlargment.
Arterial hypertension of renovascular genesis.

References:

1. A. Meystel, “Evolution of Intelligent System Architectures: What Should be Measured?”, Proceedings of the NIST Workshop on Metrics for performance and Intelligence of Intelligent Systems,” Gaithersburg, MD, 2000

2. V. Grishin, “Multivariate Data Visualization for Qualitative Model Choice in Learning System,” in Proceedings of the 1998 IEEE International Symposium on Intelligent Control, NIST, Gaithersburg, 1998, pp. 622-627

PART III
SUMMARY OF PLENARY DISCUSSIONS

Compendium of the Minutes of Plenary Discussion

Panelists: T. Balch, K. Bellman, M. Cotsaftis, P. Davis, W. J. Davis, R. Fakory, R. Finkelstein, E. Grant, J. Hernandes-Orallo, C. Joslyn, L. Reeker, E. Messina, A. Meystel, R. Murphy, C. Peterson, L. S. Phoha, Pouchard, T. Samad, A. Sanderson, A. Schultz, W. C. Stirling, G. Sukhatme, S. Wallace, A. Wild, J. Weng, T. Whalen

These notes follow the order of the papers presentation at the Workshop. Their significance is linked to the ideas and generalizations that were noticed and recorded by the panelists. Themes of these notes follow the concepts reflected in the session titles of the Workshop. Different themes generated notes of different depth and originality. Of course, this is a result of papers presented, attendees, and how stimulating the discussion was at the end of the session.

Theme 1. Features of the Industrial Intelligent Systems

The nature and embodiment of machine intelligence were discussed based upon papers:
(1) on the description of NIST ATP-funded technology development and demonstration project
(2) on the use of SOAR and CLIPS architectures to solve Towers of Hanoi and Quake II problems
(3) on the definition of Task Oriented System Intelligence

The challenge was to find the consensus among these three very different aspects of the overall problem of distinguishing salient features of industrial intelligent systems. The following statements of consensus were recorded:

1. An intelligent system was initially defined as one that works to achieve goals and to survive. (The separation of the goals belonging to different time horizon is obvious).

2. The specifics of the present situation in the area of intelligent systems is in the fact that we strive to measure system *effectiveness* and *efficiency*, not *intelligence* (primarily, because we do not have the ability to meaningfully define and measure intelligence). Also, we are under the impression that we are capable of determining the effectiveness of a system. (Questions are not usually asked about the time horizon and the scope of attention in which the effectiveness and/or efficiency are evaluated).

3. Discussion was conducted on whether intelligence is inherently embodied in hardware, not software. The consensus was reached that "Hardware is one constraint on the range of a system's admissible tasks." (A dissenting point of view: hardware is not just a constraint but rather a carrier of intelligence, or the knowledge that is required for functioning of the intelligence).

4. An intelligent system was defined as the one that exhibits flexibility, generalization, and innovations, as limited by availability of information and ability of applied algorithms.

Theme 2: Metrics and Comparison of Alternatives: Case Studies

Paper 1: Rule-based learning can be applied successfully. Rules are derived from data from simulation or experimentation with participation of a human expert. Frameworks for knowledge-based controllers provide useful platforms for alternatives comparison.

Paper 2: Knowledge extraction from raw data can be done by using visualization with a human participating. The activities of the human can be learned by the intelligent system. Using the human-computer dialog and the visual-verbal approach allows us to extract data, properties, and models. Thus, visualization can be used for intelligence testing.

Paper 3: Intelligence can be understood as the ability to make the appropriate choices, or decisions (e.g. for robots). Intelligence makes the process of choosing simpler because of the structure that is imposed upon the decision making processes. Learning can be understood as the ability to adapt to environment (i.e. at different time scales, we will have different learning processes). Use analytic hierarchy process to define weights for IQ for robots.

Paper 4: Intelligence must be measured by looking at several abilities of the system; these abilities can be integrated by using the Additive Evaluation Method for simulating the absent metric (intelligence). One of them is based upon the idea of barter exchange and boils down to transforming all evaluations to a dollar-value. Ultimately, only the human has the best sense of each value of the intelligent function.

All papers of this Theme have something in common: the human must be kept in the loop
(a) to measure intelligence, and
(b) to use metric for improving control, for analysis of tools, etc.

Also, intelligence metric is still subjective and the cost-function requires human participation for evaluating the variables and assigning the weights.

Theme 3: Measuring Performance

The following salient issues were formulated:

1. Although Life and Intelligence have many similarities, they are intrinsically different in their evaluation: we can tell what is *alive* from what is *dead*. It is much more difficult to distinguish what is *intelligent* from what is *not intelligent*.

2. Formal Requirements Specifications are needed for any system at hand. Both performance and intelligence should be evaluated against the known set of their specifications. ("Ask not how much our computers can do for us, ask what we want them to do!" and "If you can specify intelligence, I can implement it!")

3. Some of the features of intelligent systems are frequently omitted at the present time. Among them, the following should be taken into account in all cases:

   a) disambiguation,
   b) self-verification, and
   c) automated synthesis (including self-synthesis)

4. The recommended approach to the system evaluation should combine the constraint-based specification with taking into account the temporal dynamic behaviors: timed vs non-timed automata. But the hard problem is the specification: it is a part of determining the dynamic behavior, too

5. Focusing upon performance measurements can be deceptive. Indeed, the system with the best performance need not be the most intelligent. If the best performance can be pre-programmed, the effort of arriving at the system with intelligence is excessive. We need intelligence *only* if the best performance is not available otherwise.

6. Thus, measuring performance without measuring the level of intelligence is not sufficient. Focus on information measures for intelligence is required, as distinct from performance. This is why the standardized tests of performance do not say anything about future functioning of a system as an *intelligent system*.

7. It would be desirable to find a simple measure of intelligence. One of the suggested measures is: *intelligence is inversely proportional to the minimum length of description for the tasks performed by a system*.

8. On the other hand, the proper functioning of the intelligent system requires satisfaction of the optimum conditions for the subsystems that support the system of intelligence, for example: *minimize total representation size*:

   $R=\Sigma[R_m + R_a + R_r]$
   R_m – model representation
   R_a – representation of the algorithm
   R_r – representation of the residual part of the system

9. An example was discussed of a well described system that confirms the above projections: dexterous manipulation with multi-fingered robotic hand

10. The concept of minimizing the state description can be seen from the known importance of distinction between explicit (iconic) and implicit (abstract) state representations

11. Evaluation of the degree of automation is one of the factors that can help us in formalizing the way of evaluating intelligence. Successive technology generations characterized by increasing automation from a non-automated system to the autonomous system.

12. An opinion was presented that autonomy can be defined as the ability of a system to react appropriately to unforeseen situations [following its own determination of how to react]. Thus, the intelligent autonomy will be a subset of autonomy cases that leads to a success.

13. Nevertheless, one can demand for autonomous and semi-autonomous systems being evaluated on the scale (continuous) of the degree of autonomy observed.

14. Black Box Metrics was suggested in the White Paper. According to it the output Vector of performance was considered varied by the input Vector of Intelligence. Actually, that are other factors that affect the output of the black box:
    a) the number of Human Operators of Complex System
    b) the number of Loops/Operator
    c) Size of Operational Space Automated

15. A transparent Glass Box Metrics can be introduced that allows for taking into account not only the input and output but also what is going on within the box including:
    - Richness of models implemented, e.g. using Multi-models
    - Efficiency of applied algorithms, e.g. using Anytime Algorithms
    - Sophistication of planning algorithms, e.g. using Dynamic Resource Allocation

16. Among the productive examples that can be recommended for exploring the comparative importance of the Black Box and Glass Box concept: unmanned autonomous vehicles.

17. Performance Metrics for Intelligent Systems can be analyzed by formulating "Intelligence Measuring Modules" (IMM). Their calculation is based on ordered weighted aggregation operator F, and the decision maker. The basic IMM is a set $<A_1, A_2, \ldots, A_n : Q>$ where

    - A_i are relevant measurable attributes, or features of the system
    - Q are linguistic quantifiers (such as "Most", "At Least," etc.)
    - $F_{w(Q)}(A, ,\ldots, A_n) = \sum w_j b_j$
    - B_j is $_j$th best (largest) of available A_i

18. A more general metric incorporates importance factors for A_i's $<A, ,\ldots, A_n : M : Q>$

Theme 4: Modeling and Measuring Machine Intelligence

The common issue of the papers related to this Theme was to observe a "scorecard" or multiple capabilities and behaviors as characteristics of a single or multi-unit system and of the task environment are varied.

The attention was drawn to exploratory design for a community of what's important over a suite of problems. Rather than conventional sensitivity analysis varying on robot or task parameter one at a time, visualization is used to enable the researcher to discover which combination of variables matter in which circumstances.

The concept of neuromorphic architectures is concerned with systems that mimic brain architecture to implement action perceptual systems which focus their attention in a closed loop interaction with the environment, an essential feature of intelligence. Behavior of such systems is systematically studied as 2 compared with that or system with "software lesions" to see if the effect of deactivating part of the simulated brain parallels the effect of lesion to the corresponding part of a real brain.

The "metric" nature of comparison can be seen in the brain organization. Survival of the organism is too slow and admits too many alternate solutions to do the job. A possible mechanism may be task completion and minimum energy matrices driving competition between incipient sets of connections during ontogeny and learning.

A program of systematic observation and development of a robot can be a part of a natural history museum, designed to be a rich social participant in interaction with humans.

In addition to a systematic qualitative research program, quantitative metrics included who followed the robot to look at aquatic dinosaurs, how long they stayed with the robot, and how well they performed on a quiz compared with these who had not interacted with the robot.

The chief bottleneck to ride application of mobile robots is not computational speed, but interactive capability, such as vision and social intelligence.

These issues can be visualized at a different resolution level. The following four languages are underlying human behavior: DNA, brain mechanisms, natural language, and written and spoken language. Research in bio-informatics is a powerful tool for understanding the lower levels in order to achieve the goal of computing with words, in which words are the input, words are the output, and the intermediate computing remains in the background.

Some highlights can be stated as follows:

- Biological systems are inspiring, they encompass the richness of the evolutionary process (the primary research already performed by Nature).
- Simulation is increasingly feasible: as the complexity of mathematical models is growing, the futility of analytical approaches gets more explicit.
- This leads to the situation when controlled experiments are easier to conduct.
- One should not expect that all types of simulation are feasible. An apprehension was expressed concerning proposals to simulate the modular-brain concept.
- One should be very careful about metaphors used in the present days terminology. One example: Social robots are different from software agents! Awareness and expressiveness make robots social, and this can be measured or estimated. This is not the case with software agents.
- *Apparent* intelligence matters; it depends on "socialness." It is not clear presently how to judge upon this feature.
- An example can be suggested for the "socialness" evaluation: A Robot Serving as the Museum Guide.
- Man-machine interactions are primary issues in robots with "socialness" feature.
- The following list contains other factors that affect the ability to measure the level of intelligence: DNA as a part of genetic algorithms observed and/or applied, symbolic representation laws applied at each level of resolution, speech and culture of intelligent systems, physical expressions and codes of communicating intelligent systems.
- Need much more work on natural-language (NL) interfaces and computing. Eventually, the NL issues might be the key into evaluation of intelligence.

Recommendation:

1. One should systematically observe the "scorecard" of quantitative and qualitative measures of performance as one varies the capabilities of a single or multiple robot system and confronts it with a rich suite of environmental challenges (for example, groups of adults and children visiting a museum with a Robot Guide.)

2. The camouflaging of the description of processes of intelligence by gratuitous use of scientific and computational phraseology should be avoided. Let words and actions speak – keep special terminology in the background!

Theme 5: Evaluating Factors of Intelligence in Systems

The highlights of this discussion:

- Intelligence is gradual (continuous function of the features of interest) and multi-dimensional (depends of many variable factors-coordinates).

- It is preferable to assign a numerical value depending on a variable than to rank in a list that hides the dependence on the particular variables.

- The difficulty of tasks can and should be precisely measured. Thus, the evaluation of performance and intelligence might depend on prior evaluation of the "objective" complexity of tasks.

- Information-theoretical tools are especially useful for presenting the results of evaluating performance, intelligence, and the complexity of tasks..

- Factors should consider incomplete, contradictory and partially wrong information handled by intelligent systems.

- Different types of reasoning are the inherent part of the system of intelligence.

- The need for self-structuring/self-organization demonstrates itself as a component of normal learning process of the system of intelligence.

- As the process of learning develops, the system improves its own efficiency by generalizing upon similarity among multiple units of information. New, lower resolution objects emerge as a result of generalization. As this process evolves, different levels of granularity form multiresolutional hierarchies of representation.

- Standard techniques from behavioral sciences (psychology, psychometrics), biology, ecology are very useful (ANOVA, dependency analysis).

- Quantitative measures turn out to be better for efficiency of computations than qualitative/discrete ones.

- Large number of experiments are needed for Intelligent Systems if the high variance of results does not allow for forming a reliable rule.

- Sharing the results of multiple experiments is crucial for increasing the group efficiency of intelligent systems (a website and/or repository would facilitate the sharing).

- Measurement and experimentation do not provide the fully reliable value of certainty but give useful information that helps statistically the overall population of intelligent systems.

- Thus, social behavior is fundamental: it compensates for the lack of perfection of the individual intelligent system.

- Agents in a group are not totally identical, we have to find how to evaluate the optimum diversity of characteristics in the group of agents.

- There are many useful results in the intuitive approaches of the past, such as sociology, ecology, but they should be combined with contemporary information-theoretical, statistical, clustering techniques.

- Penalty-reward approach of reinforcement learning is useful for training systems as well as for measuring them without the exactly predetermined goal.

- Behavioral definitions of intelligence (Albus) can and should be put in correspondence with feature-based metrics of intelligence.

- More simple systems may behave more properly or even more "intelligently" for particular success criteria or particular environments.

Theme 6: Measuring Intelligence of Multiagent and Autonomous Networks

The major challenge for this group of intelligent systems is dealing with complexity, in particular, with exponential complexity typical for many practical cases.

Approaches:

1. Using biologically inspired systems
2. Extrasensory intelligence permissiveness
3. Metrics for embedded collaborative intelligent systems that are based on:
    - Graphical Assessment Tools
    - Various "orders" of Intelligence
    - Both applications pull and tech push
4. Domain independent measures
5. Negotiation mechanisms and coordination protocols

Theme 7: Measuring Intelligence of Distributed Systems

Four papers were presented containing a treatment of intelligent distributed systems. The following issues were highlighted:

- There is a need for highly reliable systems capable of dealing with extremely complex situations (like air traffic control…)

- These systems are typically formed of subsystems that perform specific tasks that solve some larger problem/task/or control
  – The process of decomposition is one of the key issues of analysis. An understanding should be achieved concerning the following issues: what is the principle of decomposition, how it is performed in the cases of spatial, temporal, functional, and other special cases. The possibility should be verified to aggregate the decomposed system.

– Functional aggregation of the subsystems is a separate issue because the problem of coordination emerges which should be a part of behavior generation.

- As a result of decomposition/aggregation, the problem of intelligent control can evolve: usually, it is required to modify the actions and translate these modifications into subtasks, i.e. it is required to re-optimize the system.

- The problem of optimization is resolved at the stages of planning and control. However, the system sometimes cannot implement the optimal solution. In these cases the "satisficing" contingency should be applied.

- The problem of symbol grounding has the following practical incarnation: simulating the result of planning is frequently inadequate because a lot of underrepresented information is lacking. Indeed, the Planner envisions the desirable and even probable future, but it does not affect this future: the actuators of the system that enable and activate the process do.

- Multi-resolution representation of the system should allow for evaluating the performance and intelligence at all levels of resolution.

- Multiple independent agents are different from a consolidated system with a hierarchical implementation. The rules and laws are different of applying multiresolutional methodology to multi-agent distributed systems.

- The following features are characteristic of Key Monitor Expert Systems that start from the model/role based Expert System (e.g. for Automated Monitoring):
    - Capturing Knowledge is equivalent to creation of rules; this is a difficult issue
    - Hierarchical fault tree should be carefully constructed to distinguish the branching by resolution from the branching by decision making
    - Using intelligent systems in these cases is expensive
    - It would be prudent to anticipate the human-operator resistance
    - A carefully collected information about constraints should precede the process of action selection

- The system needs supporting "Intelligent" Agents to monitor the data

- In most of the practically known cases, the intelligent system cannot capture the knowledge of experts in full detail

- Learn the optimizing strategy has limited capabilities in practice

Theme 8: Competitions: Test Beds and Metrics

1. The following observations were made:

- Test beds are good
- USAR test beds are hard to design and even harder to design performance metrics because they are so *multifaceted*
  - finding victims/perception=>victims found
  - Interface => bandwidth used (AI is not limited to full autonomy)
  - Navigation=>coverage
- Performance based metrics which take into account the number of robots collaborating (P/N) penalize multiple robots systems *(except when Illah runs the competition)*
  - Tasks factor into this, e.g., 2 robots needed to pick up heavy box

Some other non-performance metrics are costs (monetary, energy consumption, etc.) and meeting constraints during execution (e.g., formation control)

2. The following unanswered questions were detected:

- What are the metrics for mixed initiative/adjustable autonomy vs. full autonomy? [Including HCI, adaptation to drop outs]
- Does P/N really discriminate against multiple robots in *all* tasks?
  - Can we compare intelligence versus cost?
  - How do we factor control strategy?
- Are competitions inherently flawed because they don't have the right scale/scope?

Do we have any metrics/taxonomy for task complexity?

Theme 9: Measuring Intelligence of Systems with Autonomy and Mobility

Papers stressed metrics of utility, which were argued to be more useful to designers than abstract intelligence. Two task-based metrics were combined into one task determined for the process of navigation.

The architectures discussed were constructed for different goals and applications. The system developer can only evaluate a system based on his or her own goal.

Some papers were focused on the issue of graph-based searching algorithms. The goal is to optimize the creation of the graph based on the computation resource limit.

An analysis was presented, based on their work on mental development, that a fundamental criterion is not really what a machine can do in a special setting, but its capability of developing mental skills.

The works presented in this session represent well the current status of the field: there are three areas:

1. Those that address system problems: construct a system to perform some challenging tasks. Works in this category tend to use task-specific criteria. It is not always the case that the same criteria can be used for other applications, as the presenter argued.
2. Those that address a tool that can be used for many different systems. Those tools cannot be directly used in the system until a designer has done a mapping from a practical problem that he wants to solve to the tool. This kind of work concentrates on an abstraction of a particular tool from a class of problems and thus it studies an abstract tool.
3. Another direction of the work, represented by the last presentation, addressed the automation of the developmental process.

In area 1 the human is in the loop of system design, and may choose a tool in area 2 in his or her design. In area 3, the human is not in the loop of task-specific programming. Instead the human designs a program that potentially can accomplish area 1 and area 2 autonomously, at the highly developed “adult” stage.

The field has a lot of work in area 1, which has achieved some limited success. The difficulties that face us in this area are very challenging. Although area 2 can provide some useful tools for area 1, the fundamental problem in area 1 is not the problem of tools, but rather something much more fundamental: systems are task specific and thus there are no uniformly acceptable criteria at the task level. You simply use different criteria to measure performance for different tasks.

Developmental paradigm in area 3 aims at a very different dimension. Its goal is to design a system that can develop autonomously, including learning to perform many different tasks, including such as tasks that the programmer does not know at the time of programming. Then, the capability of development becomes a universal capability, independent with what tasks that the system ends up learning to perform. In other words, it is the autonomous learning capability that the area 3 is measuring, not how well the system performs each task.

If a system has a powerful capability to autonomously learn, it will do well for many tasks it learns to perform, not just for a particular task. Interestingly, human intelligence does have a uniformly accepted set of tests for different age groups. This field is called psychometrics. These tests do not test what a human child can do, but rather whether the child can learn during the test. Thus, what is tested is the autonomous learning capability.

With this autonomous learning capability, the system can learn to perform various tasks, as long as the teaching process is well designed. This new dimension is motivated by human mental development from conception time through infancy to adulthood.

Another issue is whether it is necessary for an intelligent system to learn. This is a subject that was discussed during the workshop among some participants. We seemed to reach a consensus that if the tasks are static and are easy enough to directly program, one does not have to use machine learning. However, if the environment is unknown or partially unknown at the programming time, or the environment changes significantly during the task execution, then learning is a must. Fully autonomous learning is a new dimension known as development, which enables not only machine learning, but also automation of the learning process. Since this subject is very new, the power of this new research field is yet to be demonstrated.

Theme 10: Measuring Intelligence Taking into Account Linguistical, Biological and Psychological Factors

- Many interesting ideas are being proposed related to using language and psychological testing for measuring the intelligence, but they are not sufficiently fleshed out (at least, not yet).

- Natural language encompasses much that is important in intelligence, and certain aspects of natural language processing in the intelligent systems could even indicative the degree of intelligence (though even fairly retarded people and computer equipped intelligent machines are able to learn basic human languages).

- Some of the ideas related to Natural Language were presented in terms of the Turing test, and the Turing test is certainly a test that has something to do with intelligence. However, until now we are not sure what and how this relationship works and can be interpreted. Not surprisingly, Turing Test has been criticized from a lot of points of view, and our cautious view on using it as a technique for measuring intelligence seems to be justified.

- As far as Natural Language acquisition, it was not clear whether the proponents wanted to model language development or just measuring the stage of development; the first is very hard, as all of us who are interested in modeling. However, it is clear that mere measurement of "degree of development" may not tell not much, and certainly won't help with the Turing test.

- Analysis of generalization processes by using Natural Language examples (summarization) can be considered illustrative of other algorithms of generalization working in living and computer-based creatures. It seems promising to explore similarities pf linguistic and pictorial generalization, and eventually extend it toward symbolic generalization.

Theme 11: On the aspects of Projects related to Governmental Agencies

General observations

- The amount and the diversity of issues presented at the Workshop exceed the capability of a single specialist to encompass the situation: the parable about six blind sages analyzing an elephant: some see the trunk, some the tail, some the tusks of "intelligence"

- A taxonomy of natural and artificial intelligent systems should help to illuminate (but hopefully not eliminate!) these differences in perspective

- Ask the question: "how is the measure of a specific system's intelligence actually going to be used?"

- Decompose the system into its constituent subsystems. But what if the "intelligence" is emergent at the system level?

Taxonomy of Intelligent Systems

- These are some examples of "Intelligent Systems"
    - Human
    - Dog
    - Cat
    - …
    - Mobile robot
    - Industrial manipulator
    - Process controller
    - …

- These are some Factors of "Intelligence:"
    - Sensing/perception
    - Planning/reasoning
    - Effecting/skills

- Interface/language

    - Need some sort of matrix of Factors vs
    - Types: Competencies? Requirements? …?

- **Possible uses for the measure of a specific system's intelligence…**

    - Answer the question "Can system A perform task X?"
    - Help determine where to spend R&D money
    - "Raise the bar" by establishing an "expected" level of achievement
    - Serve as an advertising bullet for an intelligent product

- **Can a System "A" Perform the Task "X"?**

- DARPA supports the research and technology development in areas where the risk is high but the payoff would be significant. One aspect of this policy is to fund generously but abandon further support if it seems as though success is unlikely. DARPA program managers are strongly urged to show meaningful evidence of progress at yearly intervals. The evidence of success for the program on intelligence could be given by demonstrating that by using this approach the reliability factors could be increased.

- In the past, the evidence of success has usually been in the form of demonstrations of utility that are sometimes of questionable value in convincing potential service users of the technology that it has utility but tend to consume a significant fraction of the allocated funds. This program can result in developing fundamental techniques for testing that would be impossible to question and give them a voluntary interpretation.

- If one or more metrics could be devised for each existing governmental program, that are:

  Directly relevant to the area being funded,
  Related to the potential for a successful outcome, and
  Measurable at reasonable cost,

  it would be easier for DARPA management to evaluate progress and potentially increase the fraction of program funding that is devoted to improvement of technology.

- Since most of the advanced governmental programs are based on or include systems that can be said to embody or include "intelligence" as part of their design, funding support for the Workshop was a logical action to take.

PART IV
DECISIONS OF ADVISORY BOARD MEETING

Decisions
of the Advisory Board Meeting conducted on August 14, 2000

1. The meeting of Advisory Board was conducted to provide an opportunity of a personal encounter and communication among the Advisory Board Members. The importance of the regular e-mail communication was noted. Members agreed to continue the effort of maintaining the Multidisciplinary Community for Intelligent Systems measurements and analysis.

2. The Board continued the discussion of the issue what should be labeled "intelligent system" and how to productively define "intelligence," since measuring something not clearly defined might not be relevant. There was a consensus that Intelligent Systems can be distinguished by their ability to

a) generalize
b) build representation
c) make choices
d) formulate goals

These abilities are demonstrated by intelligent systems in different degree and they, probably, should be used for establishing the Vector of Intelligence.

3. The ability to make choices should be regarded as a central property of intelligence. Other properties of interest are linked with intelligence, too. Yet, other properties might and probably should be considered separately from the ability of making choices, e.g. the ability to process, represent, and communicate knowledge, as well as the ability to formulate the goals and determine their own behavior.

4. The consensus was that the effort should continue to be directed towards modeling the intelligence focusing specifically upon systems that a) make choices (suitable, or appropriate ones), b) form their own goals. Formation of goals is linked closely with mechanisms of intentionality. Part of the discussion was focused upon the place of learning in the mechanisms of intelligence: whether it should be considered a separate "ability'" or it is built-in within all other abilities, e.g. as in the list in p.2 of this document.

The opinion of the Board was that (at least initially) we should be interested in Systems for Making Appropriate Choices.

5. The Board decided to coordinate the activities of the research community interested in Intelligent Systems around the Systems Capable of Making Appropriate Choices. and their critical experimental and analytical characteristics that allow for evaluation of their performance in a particular environment. Within these systems a subdomain should be recognized of systems that form their own goals. Probably, other subdomains can be delineated, too. It would be important to discover and formulate these subdomains as well as to demonstrate the relationships among them.

6. A part of the discussion was concentrated upon linkage between the concept of "success" and the concept of "choice." The concept of "success" is the actual measure of performance of the intelligent system. This measure is ingrained within the present definition of intelligence (by J. Albus). The phenomenon of "choice" seems to be the tool that serves the "ability to act appropriately." The importance of the issue of "choice" was underlined by the members of Board and the decision was made to analyze the situations where the success in not the matter of *chance* but rather the matter of *choice*.

7. A rough scale of the degrees of intelligence was agreed upon:

Degree III — Self-deciding systems
Degree II — Self-targeting systems, that implicitly incorporate their goal in their decision
Degree I — Self-deciding and self-targeting systems that are educable (W. Freeman's suggestion).

Educable systems are those that autonomously formulate the goals for their learning subsystem.

8. The Board has agreed upon the short term focus of research in the area of intelligent systems and measuring their performance and intelligence. As a result of our short term research activities, the research community should learn how to predict the IS performance if the system will be considered within a different environment (new but related to the previous one). The last focus of research was proposed by C. Weisbin, and the meeting decided to concentrate around this focus at least during the upcoming year ("Weisbin's Challenge").

9. The Board outlined how the work on Weisbin's Challenge should be initiated. Meeting decided to formulate 10-12 research problems collectively related to measuring the performance and the active characteristics of the intelligent system (their "intelligence") for the cases of systems that make appropriate choices, form their own goals, or both.

For each of these systems the preferability should be compared of two technical solutions:

a) one of them based upon a single, general purpose machine
b) another based upon utilization of multiple limited capabilities systems

The (a) and (b) would allow to understand what is preferable: to focus upon universal (broad) or specialized (narrow) types of intelligence in developing IS.

10. The Board agreed that there is an urgent need of developing a draft of the Vocabulary in the Area of Intelligent Systems. The draft of the Vocabulary should be distributed among the members of Advisory Board for collecting comments and issuing a corrected and improved version.

Among the terminological issues that demand for urgently resolving them are the following terms:

- state space
- variables
- goal
- gestalt
- autonomy
- complexity
- intentionality
- representation
- learning
- behavior.

The goal of this iterative work on the Vocabulary is to achieve a consensus within the community on how to discuss the issues in the area of Intelligent Systems.

The Board decided to distribute these decisions via e-mail and to dedicate the next meeting to the topics bounded to the solution of these problems.

APPENDICES

APPENDIX A
THE PRELIMINARY DISCUSSIONS

In this part, the excerpts from the Preliminary Discussion of the Advisory Board Members are given. Several months of active exchange helped to clarify many issues that precipitated into the Workshop Agenda and its panel discussions. The discussion was conducted by Alex Meystel.

DEFINING INTELLIGENCE

LET US CLARIFY WHAT THIS PHENOMENON IS

Dear Advisory Board Member,

As a working definition of intelligence, we use the following statement (proposed in 1991 by one of the Advisory Board members):

"Intelligence is the ability of a system to act appropriately in an uncertain environment, where appropriate action is that which increases the probability of success, and success is the achievement of behavioral subgoals that support the system's ultimate goal.

Another member of the Advisory board stated:

"We regard as "intelligent system with autonomy" only a system that can function in a self sustained manner, i.e. has information from the World of what is going on, updates its representation of the World, checks it with the goal, evaluates the situation, develops behavior that is appropriate in this situation and executes (actuates) this behavior, and again, receives the information from the world, and so on."

Finally, an opinion was voiced by the third member of the board that:

"The intelligence is incorporated in the mechanisms of inferring decisions and/or self-generating new rules from existing ones in combination with external data. These properties might exist as a potentiality, they should not be associated with really successful functioning."

In other words, we have three platforms proposed about intelligence:

No. 1: success in achieving goal means "intelligence"

No. 2: self-sustaining functioning is "intelligence"

No. 3: abilities to infer and learn mean "intelligence," nothing else matters!

What do you think about this trichotomy?

A. Meystel

NOT EVERYTHING IS ADDRESSED IN THE WHITE PAPER…

1. You didn't come anywhere near covering the spectrum of philosophical views of intelligence (just start to read the mind/body literature!) I would scale back your analogies to human intelligence and testing to something more pragmatic. Your a-y classification of measurable characteristics goes in the right direction but seems too constrained by existing systems and ways of representing information.

2. You don't talk much about learning which is a critical characteristic of intelligence. It's there, but it's primarily implicit.

3. I would define intelligence relative to a domain of application. Even in the human cases there are people who are "car intelligent" but "literature ignorant" - different domains, different abilities. Also in the human domain you have different types of intelligence (Gardner's 7, Sternberg's 3, etc.) - do you want to try something similar in the autonomous systems?

4. What's the goal of these metrics? Are they do be used in a TREC type environment?

John Cherniavsky, NSF March 16, 2000

INTELLIGENCE AND THE REQUIREMENT OF BEING SELF SUSTAINED

Interestingly enough, we regard as "intelligent system with autonomy" only a system that can function in a self sustained manner, i.e. has information from the World of what is going on, updates its representation of the World, checks it with the goal, evaluates the situation, develops behavior that is appropriate in this situation and executes (actuates) this behavior, and again, receives the information from the world, and so on.

No matter whether this is a human, a robot, a manufacturing system, an e-commerce system - it is a full cycle of activities oriented toward being "self sustained".

I wonder whether a behavior can be regarded as "intelligent" without this component of being used by a "self sustained" creature.

Maybe, this is why some of our intelligence-oriented endeavors fail: we consider just a component and do not take in account the whole creature? If this is correct, one should be suspicious of any set of criteria that are not derived from the goals and means of the overall functioning. Would you agree with this? (I hope, I am not intrusive!)

From the letter by Moderator

These are difficult and controversial questions. Which is why I like limited tests such as TREC or the DARPA speech understanding. Other researchers (Gelernter for example) dismiss the notion of intelligence (human that is) as nonsensical in machines and would feel the search for intelligence metrics as doomed from the beginning.

John Cherniavsky March 20, 2000

… These are difficult and controversial questions. Which is why I like limited tests such as TREC or the DARPA speech understanding. Other researchers (Gelernter, for example) dismiss the notion of intelligence (human that is) as nonsensical in machines and would feel the search for intelligence metrics as doomed from the beginning.

John Cherniavsky April 3, 2000

KNOWLEDGE IS NOT SUFFICIENT TO QUALIFY FOR INTELLIGENCE

Alex,

I'm a big fan of Allen Newell's definition of intelligence (see “Unified Theories of Intelligence”), which is essentially that the intelligence of a system/agent is its ability to use its available knowledge to select appropriate actions to achieve its goals.

Available knowledge is important because a system that does not have knowledge available is just ignorant, whereas a system that has knowledge but doesn't use it is stupid (not intelligent). This also provides an entry for learning because available knowledge can be construed as what should be expected to be learned from experiences with the world.

Selecting actions is critical because intelligence without action is meaningless.

Goals are critical because action without purpose is also meaningless.

There is another dimension as to the generality of the intelligence of a system/agent based on the breadth of knowledge it can acquire, encode, and use, the breadth of actions it has available, and the breadth of goals it can attempt. All of these are related to the types of environments that the agent will be successful in.

This definition fits pretty well with 1, although the emphasis in the above definition is on selecting actions that the system thinks will achieve its goals (and not on some probability). I also think that adding uncertain environments in unnecessary. Chess has no uncertainty and requires intelligence.

Autonomy is probably another dimension, but is related to the generality of intelligence - what environments the system/agent can be successful in.

Hope this helps. These are interesting issues.

John Laird April 3, 2000

FINDING THE UNIFIED TECHNIQUE WILL INCREASE EFFICIENCY OF TESTING

The problem with all these contests is that the measures are very task oriented and thus, specialized. Each system is approached individually and its individual performance is measured.

Our intention is to standardize the measures of performance of intelligent systems so that one could judge the level of intelligence of the system separately from its present concrete application. It is not as far-fetched as it might seem. Indeed, the problem of software reuse will release huge amounts of funds because we will stop developing "new" pieces of software just because the application is different. (And this is just one of many justifications for standardizing the measures of performance of intelligent systems.

Look, if we determine that the success of functioning depends on a particular set of abilities: for example, the ability to search in the large set of data, or the ability to generalize upon action rules, or upon object descriptions, or depends on other abilities, then, we will be able to recommend a particular intelligent control system, or a particular pattern discovery system, or another system which contributes to the overall intelligence, for a variety of applications, where these abilities are critical.

Consider the examples that you've mentioned in your letter: data mining software, information retrieval software, speech understanding software. Obviously, they contain reusable sub-systems. These subsystems cluster information, search for patterns, recognize the anticipated pattern, and perform several more typical tasks. These typical tasks performance could be evaluated in terms of objective measures of the particular abilities (that are components of their intelligence and determine the level of their intelligence).

But we do not know anything about their level of intelligence in terms of objective measures of the set of their abilities. All we know is how this or that software package was working with a concrete task assignment, and this does not allow to say anything how good it might be for another task assignment. Its subsystems for control, recognition, etc., might be quite dumb and I should avoid their reuse. Or they might be very powerful and I should look forward to their reuse!

As you've suggested earlier, I studied TREC results as much as they are available. This is a great work. But it does not allow me to make a judgment of how much intelligence these systems rely upon and/or demonstrate. Maybe, they are very powerful, and I can use them for dealing with large knowledge bases of the intelligent mobile vehicles, or for information processing in the autonomous vehicle computer vision system. However, I do not have this information and must invent these systems anew.

Maybe, my questions about defining intelligence are more pragmatic in their essence than it seems. Maybe, by asking you I am asking a right person. Maybe, your vision and experience will be extremely helpful in PROPER FORMULATION of these really new problems.

Alex Meystel

FOLLOWING THE BIOLOGICAL MODELS MIGHT BE HELPFUL

[From the very beginning of this discussion, clear division of participants in several clusters affected the style and the first results of the exchange. Participants belonging to different clusters had different initial premises of “how are we supposed to approach our thinking about intelligence.” The groups belonging to the “different schools of thought” were using slightly different vocabularies and probably different underlying models for describing functioning of "intelligences." Two very prominent groups can be mentioned including:

a) researchers thinking about intelligence in the terms of biological models where processes of evolution meant to be a component, and another – thinking in terms of computational intelligence

The question was:

In other words, we have three platforms about intelligence:
No. 1: success in achieving goal means "intelligence"
No. 2: self-sustaining functioning is "intelligence"
No. 3: abilities to infer and learn mean "intelligence," nothing else matters!
What do you think about this trichotomy?

Model 3 is the weakest - doesn't distinguish intelligence from the performance of any existing high-grade adaptive control system.

Model 1 is better - but it doesn't specify that a system must be enabled to create its own subgoals in the context of the ultimate goal prescribed by the agent that built it and released it, and to evaluate 'appropriateness' and its own 'success' by criteria of its own design. These functions in biointelligence are subsumed under intentionality. An intelligent device must have this.

Model 2 is best of the 3 - incorporates the action-perception cycle that characterizes biological systems, which is the mechanism of intentional action, but fails to address the complementary property of assimilation, by which organisms construct and maintain a fully integrated life-long store of information through learning through actions into the World, or the mechanisms of reafference by which biosystems determine the information that is to be taken from the World, as the basis for making their decisions.

Walter Freeman March 31, 2000

INVARIANCE OF INTELLIGENCE

DEFINING SOMETHING MIGHT BE A POWERFUL THING (SOMETIMES)

May 16, 2000

Dear Dr. Kanayama:

1. Let us try to discover what is the substance of our argument; then, we will try to address it. You quoted my understanding of the problem: defining the intelligence of the system so that you could judge how it affects functioning of the vehicles? Then, you are trying to explain that you do not see the problem of finding how the intelligence affects functioning, but rather you see the problem in making these systems function well by equipping them with a custom made intelligence.

Please, understand that the problem that I have formulated is not INSTEAD of the problem that you are solving, it is a DIFFERENT problem. The solution of architectural problem should make easier searching for a solution of your problem. You will see it from positions 2 and 3 of my letter.

This is what you wrote:

First I want to define the problem I work on. Only after then, you are able to evaluate the performance of the system as a problem solver. Possible problems for groups of autonomous unmanned vehicles could be: playing soccer, playing football, clearing a land-mine field, clearing a devastated city area by a tornado, chasing a fleeing prisoner, standing-off against a criminal with hostages, driving themselves in a row in a highway, placing themselves in a museum to watch if someone hurts the masterpieces, serving people in a reception with drinks and hors d'oeuvres, line-dancing, ballet-dancing, fighting against an enemy, and so forth.

A specific team of autonomous unmanned vehicles may be good at one of the problems, but may not be good at another.

I would like to compliment you on an impressive list of possible intelligent robots that will surround us very soon.

2. Now, let me ask you two questions:

a) Can you see that all of these proficient robots (each in its domain) will have something in common architecturally?

b) Can you imagine that one might be interested in selling the box of "system's intelligence" to all your companies that manufacture these skillful, agile, and cunning creatures?

The whole issue is hidden in your belief that any system must be its own specifications oriented, while I believe that the system's intelligence is beyond particular specifications.

It is the INVARIANCE for all these intelligent robots that can be manufactured separately, have capabilities of all of the above robots, will be more reliable and cost less.

3. I would like to remind you one important thing. About 40 years ago, the systems for control and automation of metalcutting machines were designed and manufactured individually. Depending on the skills that a particular machine was supposed to demonstrate, we were used to design very sophisticated electrical schemata, and the machines were successfully functioning.

Then, people realized that all of these machines could be controlled from the same stereotypical "intelligence" and this is how the CNC systems emerged. Programmable controllers turned out to be another solution for the problem. Now, each complicated automated machine is controlled from the same PC computer that embodies its intelligence. The technology develops as usual: from individualized custom-made solutions to a typical architecture.

The same will happen with intelligent systems as soon as we understand the nature of intelligence better.

Moderator

May 25, 2000

WHICH SIDE OF THE ARGUMENT...

Dear Advisory Board Members:

I found this question in the recent mail:

Prof,

The notion that rocks have consciousness is just as counterintuitive as your and Albus' notion that a thermostat is intelligent.

Which side of this argument are you on?

Cheers, Mike

Help me to answer this question. As a moderator, I would suggest to read Hans Moravec s letter about John Searle's review of RayKurzweil, April 8, 1999 (see http://www.frc.ri.cmu.edu/~hpm/project.archive/general.articles/1999/NYRB.990325.html).

Excerpts from this letter are given below.

Which side are you on, indeed?

Letter re. John Searle's review of Ray Kurzweil, April 8, 1999

Subject: Re: "I Married a Computer" by John R. Searle, April 8, 1999

To the Editor, New York Review of Books:

In the April 8 NYRB review of Raymond Kurzweil's new book, John Searle once again trots out his hoary "Chinese Room" argument. So doing, he illuminates a chasm between certain intuitions in traditional western Philosophy of Mind and conflicting understandings emerging from the new Sciences of Mind.

Searle's argument imagines a human who blindly follows cleverly contrived rote rules to conduct an intelligent conversation without actually understanding a word of it. To Searle the scenario illustrates machine that exhibits understanding without actually having it. To computer scientists the argument merely shows Searle is looking for understanding in the wrong places. It would take a human maybe 50,000 years of rote work and billions of scratch notes to generate each second of genuinely intelligent conversation by this means, working as a cog in a vast paper machine. The understanding the machine exhibits would obviously not be encoded in the usual places in the human's brain, as Searle would have it, but rather in the changing pattern of symbols in that paper mountain.

Searle seemingly cannot accept that real meaning can exist in mere patterns. But such attributions are essential to computer scientists and mathematicians, who daily work with mappings between different physical and symbolic structures. One day a computer memory pattern means a number, another it is a string of text or a snippet of sound or a patch of picture. When running a weather simulation it may be a pressure or a humidity, and in a robot program it may be a belief, a goal, a feeling or a state of alertness. Cognitive biologists, too, think this way as they accumulate evidence that sensations, feelings, beliefs, thoughts and other elements of consciousness are encoded as distributed patterns of activity in the nervous system. Scientifically-oriented philosophers like Daniel Dennett have built plausible theories of consciousness on the approach.

Searle is partway there in his discussion of extrinsic and intrinsic qualities, but fails to take a few additional steps that would make the situation much clearer, but reverse his conclusion. It is true that any machine can be viewed in a "mechanical" way, in terms of the interaction of its component parts. But also, as Alan Turing proposed and Searle acknowledges, a machine able to conduct an insightful *conversation, or otherwise interact in a genuinely humanlike fashion, can usefully be viewed in a "psychological" way, wherein an observer attributes thoughts, feelings, understanding and consciousness. Searle claims such attributions to a machine are merely extrinsic, and not also intrinsic as in human beings, and suggests idiosyncratically that intrinsic feelings exude in some mysterious and undefined way from the unique physical substance of human brains.*

Consider an alternative explanation for intrinsic experience. Among the psychological attributes we extrinsically attribute to people is the ability to make attributions. But with the ability to make attributions, an entity can attribute beliefs, feelings and consciousness to itself, independent of outside observers' attributions! Self-attribution is the crowning flourish gives properly constituted cognitive mechanisms, biological or electronic, an intrinsic life in their own mind's eyes. So abstract a cause for intrinsic experience may be unpalatable to classically materialist thinkers like Searle, but it feels quite natural to computer scientists. It is also supported by biological observations linking particular patterns of brain activity with subjective mental states, and is a part of Dennett's and others' theories of consciousness.

Elsewhere Hilary Putnam and Searle independently offered another kind of objection. If real thoughts, feelings, meaning and consciousness are found in special interpretations of the activity patterns of human or robot brains, wouldn't there also be interpretations that find consciousness in less traditional places, for instance (to use their examples), in the patterns of particle motion of arbitrary rocks or blackboards? Putnam, once a champion of the interpretive position, found this implication impossibly counterintuitive, and turned his back on the whole logical chain. To Searle, it simply bolsters his preexisting opinion. But counterintuitive implications do not refute an idea. The interpretations required in Putnam's and Searle's examples are too complex for us to actually muster, putting the implied beings out of our interpretive reach, thus unable to affect our everyday experience. The last chapter of my recent book "Robot: Mere Machine to Transcendent Mind" explores further implications, and uncovers no self-contradictions nor contradictions with reality as we know it. Rather, the interpretive position sheds light on mysteries like the unexpected simplicity of basic physical law. It does predict many surprises beyond our immediate observational horizons, and offends common metaphysical assumptions. But today, when millions of 3D videogame players immerse themselves in increasingly expansive and populated worlds found in very special interpretations of the particle motions of a few unimpressive-looking silicon chips, is the idea of whole worlds hidden in unexpected places still beyond the pale?
Hans Moravec, January 7, 1999

May 26, 2000

A METHODOLOGY OF MEASURING THE PERFORMANCE OF INTELLIGENT SYSTEMS

Dear Advisory Board Member:

Everybody would benefit from the insights into the problem of Performance Measures gracefully submitted to us by Dr. Larry Reeker from NIST (a Member of our Advisory Board). Interestingly enough, the recommended techniques of measuring the performance could be applied to testing most of the Intelligent Systems with the elements of Autonomy.

This is what Larry wrote in his letter:

I thought you might be interested in Teasuro's discussion of evaluation of backgammon play. I remembered he had done some, and just ran into it as I reread his paper for an entirely different reason. If you are interested in the paper, you can read it at

http://www.research.ibm.com/massive/tdl.html#h1:temporal_difference_learning

It reflects three methods that have wider applicability: contests against other programs (particularly benchmark programs), contests against humans (coupled with subjective evaluation), simulation of the outcome of decisions made.

PERFORMANCE MEASURES

There is a number of methods available to assess the quality of play of a backgammon program; each of these methods has different strengths and weaknesses. One method is automated play against a benchmark computer opponent. If the two programs can be interfaced directly to each other, and if the programs play quickly enough; then many thousands of games can be played and accurate statistics can be obtained as to how often each side wins. A higher score against the benchmark opponent can be interpreted as an overall stronger level of play. While this method is accurate for computer programs, it is hard to translate into human terms.

A second method is game play against human masters. One can get an idea of the program's strength from both the outcome statistics of the games, and from the masters' play-by-play analysis of the computer's decisions. The main problem with this method is that game play against humans is much *slower, and usually only a few dozen games can be played. Also the expert's assessment is at least partly subjective, and may not be 100% accurate.*

A third method of analysis, which is new but rapidly becoming the standard among human experts, is to analyze individual move decisions via computer rollouts. In other words, to check whether a player made the right move in a given situation, one sets up each candidate position and has a computer play out the position to completion several thousand times with different random dice sequences. The best play is assumed to be the one that produced the best outcome statistics in the rollout. Other plays giving lower equities are judged to be errors, and the seriousness of the error can be judged quantitatively by the measured loss of equity in the rollout.

In theory, there is a potential concern that the computer rollout results might not be accurate, since the program plays imperfectly. However, this apparently is not a major concern in practice. Over the last few years, many people have done extensive rollout work with a commercial program called "Expert Backgammon," a program that does not actually play at expert level but nevertheless seems to give reliable rollout results most of the time. The consensus of expert opinion is that, in most "normal" positions without too much contact, the rollout statistics of intermediate-level computer programs can be trusted for the analysis of move decisions. (They are less reliable, however, for analyzing doubling decisions.) Since TD-Gammon is such a strong program, experts are willing to trust its results virtually all the time, for both move decisions and doubling decisions. While computer rollouts are very compute-intensive (usually requiring several CPU hours to analyze one move decision), they provide a quantitative and unbiased way of measuring how well a human or computer played in a given situation.

Larry Reeker

May 27, 2000

Dear Advisory Board Members,

Attached, you will find, a document developed by a fellow Advisory Board Member, Dimitar Filev from Ford Corporation. Let me know if you have any comments.

Moderator.

AN IMPORTANT SUBSET OF INTELLIGENT SYSTEMS

This comment is focused on one special group of intelligent systems - the intelligent control systems. What makes a control system intelligent and is there a clearly defined border between intelligent and other (nonintelligent) control algorithms? The trivial answer to this question usually is *determined based on the control methodology used. Commonly, the soft computing based control algorithms (neural / fuzzy / genetic) are considered intelligent by default because of their knowledge-based content. Such a determination, however, is opposed by the control theorists who claim that modern (conventional) control methods with their strong mathematical foundations are not less intelligent than the above mentioned soft computing technologies.*

Strictly speaking, all robust control algorithms (conventional and soft) fit the first part of the definition of Albus since they are targeted to work in uncertain environment and if properly designed they generate appropriate actions to increase the probability of success with respect to a given criterion. In a broad sense, however, there are very few, if any, control algorithms that satisfy the definition of system intelligence. While evaluating the level of intelligence based on this definition (to avoid the confusion of introducing a new one) we have to take into account:

- *type of uncertain environment*
- *strategy of achieving the goals*
- *capability of the system to automatically create and update its subgoals.*

Most of the well-established methods for robust control design provide the capability to deal with small parametric and structural uncertainties and therefore include a basic level of intelligence in the control system according to the definition of Albus. Situational uncertainty, e.g. drastic changes in the environment that are due to completely different operating conditions, severe and unpredictable disturbances, etc., completely alter system dynamics, and therefore require control systems with much higher level of intelligence.

These strategies of achieving the goals that deal with analysis of the situation, selection of alternative control actions in accordance with the identified environment, and subsequent adaptation convey more than basic intelligence to the control system. In this scheme the gain scheduling, adaptive control and hierarchical control are only special cases of an intelligent control mechanism that brings in the elements of perception of situation and decision making.

The flexibility of the structures offered by the fuzzy and neural models and the natural granulation of the information that is associated with these models provide some of the basic building blocks for development of intelligent control systems. In my view, we have seen only some of the advantages of these methods over the conventional (equation based) control paradigm. So far, the gain of using these technologies comes mostly from using them as universal approximators and as tools for granulation (i.e. partitioning the space and natural decomposition of the system - typical example are the so called neuro-fuzzy systems where the fuzzy/neural model is used as a powerful tool for approximation of the plant model). Fuzzy/neural systems that are introduced in such environment are an alternative and powerful tool that enriches the control toolbox but does not automatically generate a higher level of intelligence. We are about to see more intelligent control strategies, e.g. task oriented control and *hierarchical control with dynamically created and updated subgoals when we start fully utilizing the knowledge-based content and decision making capabilities of the soft computing technologies.*

Dimitar Filev

May 27, 2000

OUR EVALUATIONS OF MACHINE INTELLIGENCE SHOULD BE COMPATIBLE WITH OUR EVALUATIONS OF HUMAN INTELLIGENCE!

Dear Advisory Board Members,

The following Hans Moravec's thoughts will be interesting for you:

[THE PROPERTY OF INTELLIGENCE IS ASSIGNED BY US]

Perhaps the most unsettling implication of this train of thought is that anything can be interpreted as possessing any abstract property, including consciousness and intelligence. Given the right playbook, the thermal jostling of the atoms in a rock can be seen as the operation of a complex, self-aware mind. How strange. Common sense screams that people have minds and rocks don't. But interpretations are often ambiguous. One day's unintelligible sounds and squiggles may become another day's meaningful thoughts if one masters a foreign language in the interim. Sometimes we exploit offbeat interpretations: an encrypted message is meaningless gibberish except when viewed through a deliberately obscure decoding. Humans have always used a modest multiplicity of interpretations, but computers widen the horizons. The first electronic computer was developed by Alan Turing to find "interesting" interpretations of wartime messages radioed by Germany to its U-boats. As our thoughts become more powerful, our repertoire of useful interpretations will grow. We can see levers and springs in animal limbs, and beauty in the aurora: our "mind children" may be able to spot fully functioning intelligences in the complex chemical goings on of plants, the dynamics of interstellar clouds, or the reverberations of cosmic radiation.

[THE CRUCIAL ROLE OF SELECTION]

There is no content or meaning without selection. The realm of all possible worlds, infinitely immense in one point of view, is vacuous in another. Imagine a book giving a detailed history of a world similar to ours. The book is written as compactly as possible: rote predictable details are left as homework for the reader. But even with maximal compression, it would be an astronomically immense tome, full of novelty and excitement. This interesting book, however, is found in "the library of all *possible books written in the Roman alphabet, arranged alphabetically"---the whole library being adequately defined by this short, boring phrase in quotes. The library as a whole has so little content that getting a book from it takes as much effort as writing the book. The library might have stacks labeled A through Z, plus a few for punctuation, each forking into similarly labeled substacks, those forking into subsubstacks, and so on indefinitely. Each branchpoint holds a book whose content is the sequence of stack letters chosen to reach it. Any book can be found in the library, but to find it the user must choose its first letter, then its second, then its third, just as one types a book by keying each subsequent letter. The book's content results entirely from the user's selections; the library has no information of its own to contribute.*

The set of all possible interpretations of any process as simulations is exactly analogous to the content of all the books in the library. In total it contains no information, yet every interesting being and story can be found within it.

[WHO PERCEIVES AND WHO INTERPRETS?]

If our world distinguishes itself from the vast unexamined (and unexaminable) majority of possible worlds through the act of self-perception and self-appreciation, just who is doing all the perceiving and appreciating? The human mind may be up to interpreting its own functioning as conscious, so rescuing itself from meaningless zombiehood, but surely we few humans and other biota---trapped on a tiny, soggy dust speck in an obscure corner, only occasionally and dimly aware of the grossest features of our immediate surroundings and immediate past---are surely insufficient to bring meaning to the whole visible universe, full of unimagined surprises, 10^40 times as massive, 10^70 times as voluminous, and 10^10 times as long-lived as ourselves. Our present appreciative ability seems more a match for the simplicity of Saturday-morning cartoons.

[NEW MODELS ENHANCE OUR ABILITY TO CREATE POSSIBLE WORLDS]

Although our eyes and arms effortlessly predict the liftability of a rock, the action of a lever, or the flight of an arrow, mechanics was deeply mysterious to those overly thoughtful ancients who pondered why stones fell, smoke rose, or the moon sailed by unperturbably. Newtonian mechanics revolutionized science by precisely formalizing the intelligence of eye and muscle, giving the Victorian era a viscerally satisfying mental grip on the physical world. In the twentieth century, this common-sense approach was gradually extended to biology and psychology. Meanwhile, physics moved beyond common sense. It had to be reworked because, it turned out, light did not fit the Newtonian framework.

In a one-two blow, intuitive notions of space, time, and reality were shattered, first by relativity, where space and time vary with perspective, then more seriously by quantum mechanics, where unobserved events dissolve into waves of alternatives. Although correctly describing everyday mechanics as well as such important features of the world as the stability of atoms and the finiteness of heat radiation, the new theories were so offensive to common sense, in concept and consequences, that they inspire persistent misunderstandings and bitter attacks to this day. The insult will get worse. General relativity, superbly accurate at large scales and masses, has not yet been reconciled with quantum mechanics, itself superbly accurate at tiny scales and huge energy concentrations. Incomplete attempts to unite them in a single theory hint at possibilities that exceed even their individual strangeness.

[COMMON SENSE OF MEASURING]

In principle, if not practice, the point of collapse can be pinpointed: before collapse, possibilities interfere like waves, creating interference patterns; after collapse, possibilities simply add in a common-sense way. Very small objects, like neutrons traveling through slits, make visible interference patterns.

Unfortunately, large, messy objects like particle detectors or observing physicists would produce interference patterns much, much finer than atoms, indistinguishable from common-sense probability distributions because they are so easily blurred by thermal jiggling.

Because, for humans, common sense is easier than quantum theory, workaday physicists take collapse to happen as soon as possible---for instance, when a particle first encounters its detector. But this “early collapse” view can have peculiar implications. It implies that the wave function can be repeatedly collapsed and uncollapsed in subtle experiments that allow measurements to be undone through deliberate cancellation at the experimenter’s whim.

Einstein was troubled by the implications of quantum mechanics, and he devised thought experiments with outcomes so counterintuitive he felt they discredited the theory. Those counterintuitive outcomes are now observed in laboratories and utilized in experimental quantum computers and cryptographic signaling systems. Soon, more advanced quantum computers will allow the results of entire long computations to be undone.

Common sense screams that measurements are real when they register in the experimenter’s consciousness. This thinking has led some philosophically inclined physicists to suggest that consciousness itself is the mysterious wave-collapsing process that quantum theory fails to identify. But even consciousness is insufficient to cause collapse in the thought experiment known as “ Wigner’s Friend.” Like the more famous “ Schr dinger’s Cat,” Wigner’s friend is sealed in a perfectly isolating enclosure with a physics experiment that has two possible outcomes. The friend observes the experiment and notes the outcome mentally. Outside the leakproof enclosure, Wigner can only describe his friend’s mental state as the superposition of the alternatives. In principle these alternatives should interfere, so that when the enclosure is opened one or another outcome may be favored, depending on the precise time of opening. Wigner might then conclude that his own consciousness triggered the collapse when the enclosure was opened, but his friend’s earlier observation had left it uncollapsed.

[MANY-WORLDS INTERPRETATIONS]

In a 1957 Ph.D. thesis, Hugh Everett gave a new answer to that question. Given a universally evolving wave function, where the configuration of a measuring apparatus, no less than of a particle, spreads wavelike through its space of possibilities, he showed that if two instruments recorded the same event, the overall wave function had maximum magnitude for situations where the records concurred and canceled where they disagreed. Thus, a peak in the combined wave represents a possibility where, for instance, an instrument, an experimenter's memory, and the marks in a notebook agree on where a particle alighted---eminent common sense. But the whole wave function contains many such peaks, each representing a consensus on a different outcome. Everett had shown that quantum mechanics, stripped of problematical collapsing wave functions, still predicts common-sense worlds---only many, many of them, all slightly different. The "no-collapse" view became known as the "many-worlds" interpretation of quantum mechanics.

[INTELLIGENCE DETECTS SINGULARITIES WITHIN THE CHAOS]

No complete theory yet explains our existence and experiences, but there are hints. Tiny universes simulated in today's computers are often characterized by adjustable rules governing the interaction of neighboring regions. If the interactions are made very weak, the simulations quickly freeze to bland uniformity; if they are very strong, the simulated space may seethe intensely in a chaotic boil. Between the extremes is a narrow "edge of chaos" with enough action to form interesting structures, and enough peace to let them persist and interact. Often such borderline universes can contain structures that use stored information to construct other things, including perfect or imperfect copies of themselves, thus supporting Darwinian evolution of complexity. If physics itself offers a spectrum of interaction intensities, it is no surprise that we find ourselves operating at the liquid boundary of chaos, for we could not function, nor have evolved, in motionless ice nor formless fire.

[INTELLIGENCE DEVELOPS THE IMAGES OF POSSIBLE WORLDS]

The similarity between Everett's "many worlds" and the philosophical "possible worlds" may become stronger yet. In "many worlds" quantum mechanics, physical constants, among other things, have fixed values. Gravity, in objects like black holes, loosens the rules, and a full quantum theory of gravity may predict possible worlds far exceeding Everett's range---and who knows what potent subtleties lie even further on? It may turn out, as we claw our way out through onion layers of interpretation, that physics places fewer and fewer constraints on the nature of things. The regularities we observe may be merely a self-reflection: we must perceive the world as compatible with our own existence---with a strong arrow of time, dependable probabilities, where complexity can evolve and persist, where experiences can accumulate in reliable memories, and the results of actions are predictable. Our mind children, able to manipulate their own substance and structure at the finest levels, will probably greatly transcend our narrow notions of what is.

[WE ARE SELF-INTERPRETING BOTH OURSELVES AND THE EXTERNAL WORLD]

Like organisms evolved in gentle tide pools, who migrate to freezing oceans or steaming jungles by developing metabolisms, mechanisms, and behaviors workable in those harsher and vaster environments, ***our descendants may develop means to venture far from the comfortable realms we consider reality into arbitrarily strange volumes of the all-possible library.*** *Their techniques will be as meaningless to us as bicycles are to fish, but perhaps we can stretch our common-sense-hobbled imaginations enough to peer a short distance into this odd territory. Physical quantities like the speed of light, the attraction of electric charges, and the strength of gravity are, for us, the unchanging foundation on which everything is built. But if our existence is a product of self-interpretation in the space of all possible worlds, this stability may simply reflect the delicacy of our own construction---our biochemistry malfunctions in worlds where the physical constants vary, and we would cease to be there. Thus, we always find ourselves in a world where the constants are just what is needed to keep us functioning. For the same reason, we find the rules have held steady over a long period, so evolution could accumulate our many intricate, interlocking internal mechanisms.*

Our engineered descendants will be more flexible. Perhaps mind-hosting bodies can be constructed that are adjustable for small changes in, say, the speed of light. An individual who installed itself in such a body, and then adjusted it for a slightly higher lightspeed, should then find itself in a physical universe appropriately altered, since it could then exist in no other. It would be a one-way trip. Acquaintances in old-style bodies would be seen to die---among fireworks everywhere, as formerly stable atoms and compounds disintegrated. Turning the tuning knob back would not restore the lost continuity of life and substance. Back in the old universe everything would be normal, only the acquaintances would witness an odd "suicide by tuning knob." Such irreversible partings of the way occur elsewhere in physics. The many-worlds interpretation calls for them, subtly, at every recorded observation. General relativity offers dramatic "event horizons": an observer falling into a black hole sees a previously inaccessible universe ahead at the instant she permanently loses the ability to signal friends left outside.

See more in URL:

http://www.frc.ri.cmu.edu/~hpm/project.archive/general.articles/1998/SimConEx.98.html

Be patient to get to the last quarter of THIS DOCUMENT. It raises important issues concerning world representation. As you know, H. Moravec was one of the first creators of vehicles with elements of autonomy.

Moderator

May 29, 2000

Dear Advisory Board Member,

Dr. Eric Horvitz (a member of our Board) has suggested to distribute the attached paper among the Advisory Board Members. This paper is related to dealing with uncertainties as a part of the process of decision making when the imprecisely computed Metrics are used.

Since any introduction of Metrics is linked with determining preferences under uncertainty, it would be meaningful to be prepared to the non-trivial situations of decision making using any proposed
Metrics.

It would be presumptuous to expect that our University education has prepared us to these situations exhaustively (partially? maybe).

I found this paper enlightening (to the extent I could understand it). All of you are expected to achieve some level of understanding of the peculiarity linked with introduction and use of Metrics for Decision Making.

Moderator

May 30, 2000

THOUGHTS AFTER READING THE WHITE PAPER

BY TOM WHALEN

The "white paper" draft seems to me to be a real gold mine, but like any mine it requires digging, sifting, and refining.

1. INTELLIGENCE AND AUTONOMY

I really like the idea of the autonomous climate control system being "motivated" to increase its autonomy by reducing the need for human intervention. (p.3) I think this could be the kernel for a better definition of what is meant by intelligence in autonomous constructed system.

Here's my stab at some global definitions:

Def.1

"A constructed system is autonomous if there is a likelihood that circumstances will arise in which no-one can predict in advance what it will do. This need not imply randomness, just complexity."

[This probably can imply both: complexity AND randomness. Moderator.]

Def.2

"An autonomous constructed system is intelligent if we can be reasonably confident that whatever unpredictable thing it does do will be something that tends toward success in the goals for which the system was constructed in the first place."

2. HOW TO FALSIFY THESE DEFINITIONS?

Thus, a claim that a system is autonomous and intelligent can be falsified in two ways: showing it is not autonomous by predicting all of its behavior in advance, or showing it is not intelligent by demonstrating that its behavior is stupid.

What an end user wants is a system that is trustworthy. If all behavior can be specified in advance, there is no need for autonomy; the intelligence and autonomy reside in the design team and not in the delivered system. If behavior can't be prespecified, then intelligence is necessary for trustworthiness; if it is lacking, the system needs to be monitored by a human operator and thus, again, lacks autonomy.

Note the statement on page of the White Paper that an intelligent system was "designed by humans (engineers and programmers)" is not true in machines that learn and self-organize except in a broadened sense of the word "designed." Even very large and complex programs that have no learning or self-organizing features need to be studied in much the same way we study social phenomena like economics or natural phenomena like weather, since no one person will ever know the program in its entirety. (For example, the Windows operating system.)

The paper has the beginnings of a structure for measuring the components of machine intelligence based on the six-box semiotic loop, but it's not very consistent.

3. MIND-BODY PROBLEM

The discussion of the "mind-body" problem crops up several times in the White Paper; I suggest making it specific by assigning perception, knowledge, and decision (behavior generation) as "mind" and assigning sensation and actuation as body. The sixth box, "world," is not part of the constructed autonomous system.

[Here, we should think twice: should the work-piece that we drill be considered a part of the drilling machine or not? The part of the world that immediately interacts with a system under consideration might be legitimately considered a part of this system. When we write the equations of the system, the "torque on the shaft" is a part of the equations of the system. Moderator.]

4. ABOUT CHINESE ROOM

Note: my personal view of the Chinese room is that performing the task without understanding Chinese is not in principle impossible, but the number of rules that would have to be written ahead of time and searched in real time by the occupant of the room is far beyond the trillions. Learning to understand Chinese is a much easier task, already mastered by over a billion people!

This is quite relevant to the issue of intelligent autonomous machines; there are tasks for which it may be within our grasp to produce a successful machine without autonomy, but it is actually easier to achieve the same level of success using an intelligent autonomous machine. A simple and very familiar example is the inverted pendulum, which is quite challenging to do with differential equations but a beginner's exercise to do with fuzzy control.

[I would like to emphasize this significant Tom's statement by capitalizing: THERE ARE TASKS FOR WHICH IT MAY BE WITHIN OUR GRASP TO PRODUCE A SUCCESSFUL MACHINE WITHOUT AUTONOMY, BUT IT IS ACTUALLY EASIER TO ACHIEVE THE SAME LEVEL OF SUCCESS USING AN INTELLIGENT AUTONOMOUS MACHINE. Moderator]

5. WHAT DOES IT MEAN TO BE "INTELLIGENT"

My initial impression is that while human intelligence testing does rely heavily on response time, my online thesaurus lists the following synonyms for "intelligent" (see the White Paper). Only three of the sixteen involve the idea of "quick-witted." Machines routinely do almost anything they can do at all more quickly than humans can do them, and they also are incapable of doing at all many things humans do quickly. There is only a small middle ground of things machines do roughly as fast as humans or that they do well but more slowly than humans.

I'm more interested in the prospects for machines that are canny. percipient, perspicacious, astute, and discerning.

Tom s insights are great!
Moderator.

May 30, 2000

THOUGHTS AFTER READING THE WHITE PAPER

by Sukhan Lee

I would like to congratulate (...) the effort to formalize the intelligent system research by establishing the measure of system intelligence. Establishing the measure (...) should not only be able to turn the intelligent system into a formal academic discipline but also provide a means of designing better and more powerful intelligent systems in practice.

I am generally impressed by the breadth as well as depth of [the proposed] measure of intelligence for a constructed system with autonomy. (...) various aspects of intelligence [are considered] including the need to learn as well as to generalize by an intelligent system. (...) a list of system specifications [are proposed] as well as the vector of intelligence as features representing intelligent functions of a system.

The list of features presented is very comprehensive. However, it is not clear how the measure of intelligence can be formulated out of such a list or a vector. Too many functional features may obscure the essence of how intelligence is generated, as they may not represent the engine but the expressions.

Having said that, I would like to pay attention to the following questions:

1) Should the intelligence measure be goal-dependent or goal-independent?

2) Should the intelligence measure be time varying or time-invariant?

3) Should the intelligence measure be resource-dependent or resource-independent?

1) (...) a question [emerges] whether there exists a universal measure of system intelligence such that the intelligence of a system can be compared independently of the given goals. A goal-independent measure may be more difficult to define, (if not impossible), and [it will be] more controversial.

A goal-dependent measure, however abstract the goal may be, can allow [for a] clear comparison among the systems of different architecture but with the same goal. For instance, for the latter case, an intelligence can be represented as how efficiently, and how optimally a system reaches the given goal by itself, i.e., the power of automatically solving problems defined as the discrepancy between the goal and the current state.

2) (...) [We should decide whether] the intelligence measure of a system should solely be based on problem-solving capability at time t or it should contain the potential increase of problem-solving capability in the future based on learning. My opinion is that we need both. But, it is better to define the two separately before integrating them together into one measure.

3) (...) [Finally, it is an important] issue whether the resources required for building systems and system operation should play a role for defining the measure of intelligence. As mentioned above, the efficiency in problem solving, I think, should be included in the measure: for instance, the time and energy required to reach a solution should be taken into consideration together with the optimality of the solution. But, I am not sure whether we should or should not include the cost of building a system.

[I WOULD APPRECIATE SENDING TO ME YOUR THOUGHTS ABOUT THE INTERESTING POINTS INDICATED BY S. LEE. Moderator]

May 31, 2000

THOUGHTS AFTER READING THE WHITE PAPER: C. WEISBIN S QUESTIONS

Would it be appropriate for you to specify for the workshop a SMALL number (~5) questions which the workshop (perhaps within working groups, or abstracted from position papers) would try to answer with some degree of specificity? The field is so broad and the interests are so varied that I am puzzled how (whether?) tangible conclusions will emerge? The field is so broad and the interests are so varied that I am puzzled how (whether?) tangible conclusions will emerge?

I WOULD LIKE TO DISCUSS WITH ALL MEMBERS OF THE ADVISORY BOARD THE LIST OF QUESTIONS THAT I PROPOSED IN THE RESPONSE LETTER TO C. WEISBIN.
LET ME KNOW WHAT DO YOU THINK, THIS IS VERY IMPORTANT:

This is the list of questions that the Workshop will try to answer:

1. What is the vector of intelligence (VI) that should be measured and possibly used as a metric for systems comparison?

2. Should VI be measured in addition, or instead of measuring the vector of performance (VP) determined by the regular specifications?

3. If two systems have the same VP, what is implied by the difference in their VI values? Can this difference be represented in $ units?

4. Is it possible (and meaningful) to have different VI measures: a) goal-invariant, b) resource-invariant, c) time-invariant?

5. What should be recommended as a test of VI and how to normalize VP so that comparison be performed at the same normalized value of VP.

These are the five questions that you have asked about. As a reminder, I would like to formulate a set of other questions that are ingrained (directly, or indirectly) in the main five questions:

6. These are the less profound ("secondary") questions that should be addressed at the workshop and possibly unequivocally answered:

a) how to form VI for various architectures?

b) should the questions 1 through 5 be related to intelligent systems, or autonomous systems, or both?

c) what is the protocol of dealing with uncertainty when the uncertain metric is to be applied in the procedures of decision making? for example how the uncertainty of planning affects the cost of goal achievement?

d) what are the guidelines in constructing the world model and determining its scope in the variety of applications? how the scope of "world model" affects the sophistication of intelligent behavior?

e) how are the questions 1 through 5 related (and the answers applied to) the systems that are working under a hierarchy of goals.

f) should a competition between intelligent systems be considered a valid method of judging VI value?

Moderator

May 31, 200

SOME COMMENTS ON THE PROBLEM OF METRICS OF INTELLIGENCE

GEORGE BEKEY HAS PROPOSED THE FOLLOWING:

1. The selection of benchmark problems on which to measure the degree of autonomy and intelligence of a system, and

2. Something orthogonal to the previous discussions: A discussion of the moral and ethical implications of building increasingly intelligent systems.

For more details on these issues see his attachment. *Moderator*

Attachment: G. Bekey s comments:

1. Benchmarks

I am a strong believer in simplicity, so my definitions and metrics are very simple. I believe that the fundamental attributes of intelligence involve:

- *Ability to perform tasks in unstructured environments*
- *Ability to learn from experience*
- *Ability to transfer knowledge from one domain to another*
- *Ability to solve complex problems, requiring deductive and inductive reasoning*

(While stated differently, these issues are similar to Jim Albus s definitions). I suggest that the following simple measures can be used as metrics for such abilities in machines:

1. *Size and complexity of programs required*
2. *Memory requirement*
3. *Solution time*

Clearly, such measures are useful only if (a) they are applied to benchmark problems, (b) all contestants use the same type and model of computer, and (c) all programs are written by comparably competent programmers, so that the programs are optimal in some sense.

Given these constraints, we could test intelligent systems A and B on the same benchmarks. The one that accomplishes the task more quickly, and does so with the least complex programs and least memory will be declared more intelligent . What could be simpler than that?

If anyone agrees with me, I would be interested in leading a discussion on the selection of benchmarks on which we can all test our systems.

(There are several hidden questions here. One of them is the question of emergent behaviors. As a system learns more and more, and is able to transfer knowledge to other domains, and solve increasingly complex problems, it may begin to do so in totally unexpected ways. The emergence of such new behaviors will be appear in its ability to solve problems more rapidly, but will not be directly measurable).

2. On the ethics of building intelligent machines

Several recent books have dealt with this subject, such as the latest books from Hans Moravec or Ray Kurzweil (The age of spiritual machines). Kurzweil predicts that by 2025 computers will be more intelligent than people (but does nor provide metrics to measure this result!). Perhaps the most thoughtful analysis was published by Bill Joy (the chief scientist of Sun Microsystems), in the April 2000 issue of WIRED. In an article entitled Why the future doesn t need us , he speculates that developments in robotics, nanotechnology and genetic engineering will inevitably lead to self-reproducing machines with increasing intelligence, whose behavior will be not only unpredictable but uncontrollable. Such machines may find human beings largely superfluous.

I would be interested in presenting a position paper summarizing current thinking on this matter and leading a discussion. It seems to me that if we are concerned only with measuring the intelligence of machines without any concern for the social implications of such intelligence, we are not fulfilling our responsibility to society.

George Bekey

June 3, 2000

WINTER S CONJECTURE

(FROM MY CONVERSATION WITH VICTOR WINTER, MEMBER OF THE ADVISORY BOARD)

Victor Winter (V. W.):

The ability to learn is generally considered a "sophisticated" behavior. Given this ability, a machine can positively change its behavior over time thereby exceeding the sum of its initial input. By this metric, a more a sophisticated system can "do more" with less initial input than an unsophisticated system.

Alex Meystel (A. M.):

I believe that intelligent system should learn. By I am hesitant to say that anything that can learn is necessarily intelligent. Having an increased degree of sophistication? Probably!

V. W.

For example, in theory, a theorem prover can discover all of mathematics from its axiomatization. Of course there are some severe limitations to this in practice.

A. M.

I have already addressed this issue in the previous part of our conversation. No, a simple set of axioms is insufficient to prove these two theorems that I have mentioned above. (I worked this out with students).

V. W.

Nevertheless, we can think of an axiomatization of a closed system as a very compact model of that system. (We are talking about the need to supply knowledge of the initial general laws that pertain to a concrete environment; we call them "axioms").

A. M.

Compact - maybe. But sufficient - hardly.

V. W.

To the person writing the axioms the system need not be fully understood (e.g., knowledge of all theorems in mathematics is not required).

A. M.

This is an unwanted surprise: your definition of "understood" means "having known all theorems related." I think that as a label, you can use any word you want. But in reality "understood" means much more, for example: "having future behaviors anticipated".

(The term "theorems" should be understood as the provable rules that hold in the environment of the subsequent functioning).

V. W.

However, this compact representation of the system can be utilized by a sophisticated machine to solve a large collection of problems--problems that may not even have been initially foreseen.

A. M.

You are talking about "sophisticated machine" that has not been defined. However even if your "sophisticated machine" is equivalent "human intelligence", I doubt that having just compact representation (you have defined it as based only upon the axioms) we will be able to solve the unforeseen problems.

V. W.

An application of this idea is in a space system where there may be multiple ways to achieve a certain behavior (e.g., pitch or rotation of a spacecraft). If one possibility fails due to a malfunction of a component another possibility can be discovered -- provided a sufficiently complete system model has been given. In this example, a space bound system can be equipped with a computer that has

(1) a non-algorithmic specification of the behavior of the system, and

(2) an axiomatization of that system.

A sophisticated system would then be able to satisfy its specification whenever possible, even in the presence of unforeseen circumstances.

A. M.

We have to add the word ALL into your last sentence; then my refutation will be stronger: not "even in the presence of ALL unforeseen circumstances."

The reason I insist on this "ALL" is because you allowed for only limited variability in the space of searching the strategies of solution.

There is a threshold, a level of variability that is required to cover the space of future behaviors sufficient for, say, survival with a definite probability.

V. W.

The presence of such a specification would to some extent address the concerns/issues raised by Searle (as described in your White Paper). For example, a robotic system having a defective (e.g., bent) arm would be able to sense (through "sensors") that the behavior of the arm - in its current state -- is not satisfying the specification. Alternate means of satisfying the specification could then be generated and considered.

A. M.

Of course, many realistic cases of robots functioning in Space presently do not require and do not exercise any sophistication not to speak of intelligence.

V. W.

The notion of learning can be taken one step further and one can consider a system, which has the ability to dynamically generate axioms.

A. M.

I agree with you ecstatically.

V. W.

Given what I have mentioned above, I believe that an interesting metric would be the "size" of knowledge required or the amount of information that must initially be provided to a system in order for it to demonstrate a certain behavior -- the smaller is the required initial information, the more *"sophisticated" the system is. These kinds of systems (if practical) could effectively utilize the increases in computing power that will occur in the future.*

A. M.

Victor, this is a great idea. I will call it Winter's Conjecture, and will circulate it among our Advisory Board Members. Prepare yourself for proving mathematically a number of theorems that entail your hypothesis. For example, the fundamental theorem that you should introduce is related to evaluation of the "amount" of this information based upon complexity of the system that is supposed to be equipped by "intelligence."

This theorem is on you. I would suggest to use a measure of variability of this minimal information and evaluate the probability of success in applying this minimum of information with a given measure of variability.

CLASSES OF INTELLIGENT SYSTEMS

June 3, 2000

Dear Advisory Board Member,

Attached, you will find, an important contribution by Steve Grossberg (see his paper in the next part of this book). This view focuses upon CONSTRAINTS that will undoubtedly make more difficult our road toward Metrics for Intelligent Systems in a particular class of intelligent systems.

The abstract also reminds about interesting problems that emerge in autonomous intelligent systems and make them different from the classical control systems. I would suggest to compare the class of systems that is described in S. Grossberg's paper and the class of systems that is described in D. Filev's statement (see my message on May 27).

Both classes belong to super-class of "intelligent systems."

But look how different they are!

I would not be surprised if their Metrics will have the same properties: belonging to the same super-class but very different.

Yours,

Alex Meystel

June 5, 2000

Dear Advisory Board Members,

I RECEIVED AN INTERESTING SET OF ANSWERS AND I WANT TO FAMILIARIZE YOU WITH THEM.

THIS LETTER I RECEIVED FROM WALTER FREEMAN, A MEMBER OF THE ADVISORY BOARD. IT CONTAINS SEVERAL FAR REACHING SUGGESTIONS, IMPORTANT FOR ALL OF US.

FIRST QUESTION:

1. What is the vector of intelligence (VI) that should be measured and possibly used as a metric for systems comparison?

W. FREEMAN S ANSWER:

In my view, we don't measure intelligence; we infer it from measurements of performance.

[ACTUALLY, ANYTHING MEASURED IS CONCEPTUALLY INTRODUCED IN THE BEGINNING. IN OTHER WORDS, ALL THINGS THAT WE MEASURE WE INFER FROM MEASUREMENTS OF OTHER RELATED THINGS. WALTER IS MORE INTERESTED IN THE QUALITATIVE CHARACTERIZATION OF INTELLIGENCE AND HE ASKS:]

W. F. A better question is: What kinds of intelligence do we propose to emulate?

MY ANSWER TO THIS QUESTION IS:

I doubt that there are different kinds of intelligence. I know that multiple intelligences is a faith (very similar to the polytheism of ancient people). It is easy to declare any manifestation of perceptual and cognitive activity to be an intelligence. It is more difficult to find what do they have in common, maybe, absolutely the same.

We all can easily demonstrate that all known phenomena of intellect and intelligence are linked with a limited number of special computational algorithms including

--combining N-tuples

--searching

--focusing attention

--grouping (which includes "combining tuples")

--evaluating and ranking the results of grouping.

SECOND QUESTION: 2. Should VI be measured in addition, or instead of measuring the vector of performance (VP) determined by the regular specifications?

W. FREEMAN RESPONDS:

Hence, what kinds of performance do we choose as benchmarks for measurement?

THIRD QUESTION: 3. If two systems have the same VP, what is implied by the difference in their VI values? Can this difference be represented in $ units?

W. FREEMAN SUGGESTS:

Instead: How might we choose and construct benchmark tasks with graded difficulty, so that we can establish the competence of new systems and then challenge them with tasks of increasing complexity.

[I THINK, THIS IS AN EXCELLENT SUGGESTION]

FOURTH QUESTION: 4. Is it possible (and meaningful) to have different VI measures: a) goal-invariant, b) resource-invariant, c) time-invariant?

W. FREEMAN PROPOSES:

In that there are different kinds of intelligence, it follows that there are different VPs. For VI, in analogy to IQ, I would suggest using a ratio of performance VP to cost of construction of a system in time and money VC, giving a scalar value that will assign due place to building a machine gun to kill a fly:

VI[n] = VP[n]/VC[n], n = 1,N [number of classes of benchmark]

[I THINK THAT THIS PROPOSAL IS MEANINGFUL: NORMALIZING IS A PROPER

THING TO DO. BUT I DISAGREE TO CONSIDER THESE DIFFERENT MODES OF EVALUATION OF ONE INTELLIGENCE TO BE DIFFERENT INTELLIGENCES. WHY?]

FIFTH QUESTION: 5. What should be recommended as a test of VI and how to normalize VP so that comparison be performed at the same normalized value of VP.

WALTER FREEMAN EXPLAINS AND SUGGESTS THREE BENCHMARKS:

This question would be answered under #3 and #4. I suggest N = 3 benchmarks, relating to comprehension through perception, planning action, and dynamic reasoning through decision:

n = 1. Pattern classification - for example, detection of chemical explosives of increasing variety in increasingly complex background odorant environments. We have excellent chromatographs, but there is a need here for the artificial dog behind the artificial nose.

n = 2. Spatial navigation - for example, foraging for fuel in natural environments of graded complexity. Gray Walter's autonomous tortoises are still [in my opinion] the best of breed in this respect.

n = 3. Comprehension of instructions in a natural language — for example, accomplishing sequences of operations, each conditional on the steps preceding. Ross Ashby's 'homeostat' might offer a suitable early benchmark.

[I THINK THAT THIS GIVES AN ANSWER TO MANY QUESTIONS RELATED TO METRICS OF INTELLIGENCE]

WALTER BELIEVES THAT THIS MY SIXTH QUESTION IS PREMATURE, THAT WE ARE NOT AT THE STAGE THAT QUESTION No. 6 COULD BE APPROACHED RIGHT NOW. WELL, I DISAGREE.

Moderator

EMPHASES IN RESEARCH OF BUILDING INTELLIGENT MACHINES AND MEASURING THEIR INTELLIGENCE: THE ISSUE OF ETHICS

by J. Albus

June 6, 2000

I agree with George Bekey that we should spend some time discussing the ethics of building intelligent machines. I feel that the concerns voiced by Bill Joy should not go unanswered, and the predictions of Moravec and Kurzweil should not remain unchallenged.

I, for one, feel that the most important characteristic of intelligent machines is that they have the capacity to perform useful work, i.e., to create wealth. I also fervently believe that the biggest problem in the world today is poverty, i.e., the lack of wealth. (I see poverty is the fundamental cause of hunger, disease, ignorance, pollution, intolerance, oppression, lack of medical care, and lack of education.) I therefore would argue that we should focus on how to make it possible for intelligent machines to eliminate poverty world wide within 50 years. In my opinion, to do anything else is unethical.

As for the concerns of Joy, I believe they are largely unfounded and overblown, and the predictions of Kurtzweil, and Moravec are for the most part wildly exaggerated. However, these kinds of sensational concerns fan the flames of the Frankenstein myth in the popular imagination and thereby create a major distraction. They divert attention from real problems that intelligent machines could solve by inciting fears of scenarios that are highly improbable. And they divert attention from what realistically could be done to alleviate human suffering in the near future.

Jim Albus

June 6, 2000

MODERATOR S RESPONSE:

I support Jim's letter emphasizing the utter importance of ethics in the area of Intelligent Systems. I also agree that one of biggest problems in the world is "lack of wealth."

However, there are two kinds of wealth: Material and Intellectual ones. Intellectual wealth is almost always an asset especially for those in poverty, while material wealth can't frequently help even those who is rich intellectually. Material wealth is not a universal remedy. While curing known problems it creates new. It is a mixed blessing.

I consider intelligent systems to be our helpers not in creation of material wealth as the first priority, but in elimination of Ignorance. Ignorance produces and maintains poverty both material and intellectual. Ignorance maintains unethical environment. Ignorance is the major adversary of Intelligence, so it becomes an adversary of the carriers of Intelligence.

Material wealth is frequently created by developing sophisticated and sometimes even elegant but unethical methods (including algorithms and even software packages) whose only goal is "to separate a client with his/her money." Intelligent Systems can become a powerful tool of this unethical process. It is especially terrible to be robbed by an Intelligent System tuned to create somebody's material wealth in an unethical way.

Sometimes, we give up on students that cannot figure out how things are associated with each other, and graduate them anyway -- this is when we make a step down even if we succeeded in helping them to receive a position with a major insurance company developing a huge material wealth. And this is also unethical, too.

We must think ethics before we construct anything intelligent. We should ask: hey, what is this for? whose material wealth will it increase?

But understanding how the intelligence works -- we must in all cases! This is why the analysis of Metrics for Intelligence should lead us in the right direction unmistakably.

But as far as applying this Intelligence in practice, we should ask firmly: hey, what is this for?

(Unless they offer us very good money...)

Moderator

WE DO NOT NEED JUST *ANY* KIND OF INTELLIGENT SYSTEM!

By John C. Cherniavsky

I'll comment on this though it's a bit far out of the "measuring machine intelligence" category.

First Joy is just raising questions that ethically should be raised by all scientists in the performance of their research. During a recent DARPA/NSF talk he fully acknowledged the benefits of NGR (Nanotechnology, Genetics, Robotics - I'm not sure I've got it in his order) in feeding the hungry, increasing llifespan, and generating general welfare for all peoples. Contrary to what Jim asserts, he is not ignoring the benefits of NGR. He is pointing out potential dangers. His main immediate concern, rightfully so in my opinion, is the possible creation of biological and/or nano-biological organisms that replicate and that have no natural controls on their replication. He is concerned that there are no regulatory bodies that oversee this sort of research which he sees as possible in the next 10 years or so. He is also very concerned about the possible low cost of entry for this research leading to possible inexpensive terrorism on an unprecedented scale (eg. the atomic bomb built in the garage scenario).

We certainly have regulatory bodies overseeing research on biological warfare agents, yet none on similar research that could accidentily (or deliberately) release replicating organisms into the environment. How concerned you are depends upon your view of how likely this is to occur and how likely it is to be low cost and easy, but it should be thought about and not dismissed out of hand. This particular danger has very little to do with intelligence and controls on research could be very similar to controls on research for other potentially harmful biologicals (Ebola virus for example) with a twist that these controls won't work if the technology becomes too cheap, too easy, and too ubiquitous.

Joy's concern about intelligent robots is more long term and speculative and based a lot on Kurzweil and Moravec's writings. He again advocates serious study and perhaps safety controls on research by oversight scientific bodies. If you don't believe that a truly intelligent robot will be built, then of course you have no concerns.

If you do believe that such robots will be built, you should be concerned about rights for that robot - after all it's intelligent - and the possibility that such robots would pose dangers. Long term speculation true - but not too early to be discussing in an open forum which is just what Joy, Kurzweil, and Moravec are doing.

Will these sorts of concerns chill research? Quite possibly. Just look at the fairly benign research on genetically altered crops and the furor in the European Community. Is that necessarily bad? Again it's a question of risk perception. All of society should be involved in such debates, not just the knowledgeable scientists and the cost/benefits of such work be fully debated.

After all, there is another solution to world poverty and that is reducing the world's population. Maybe we don't need Moravec style robots.

John C. Cherniavsky, Ph.D.
Senior Advisor for Research, EHR
National Science Foundation *June 7, 2000*

June 8, 2000

WHAT MEASURES INTELLIGENCE: A COMPETITION OR A SPECIAL TEST?

A MEMBER OF ADVISORY BOARD R. GARNER SAYS:

> *The question was: f) should a competition between intelligent systems be considered a valid method of judging VI value?*
>
> *I would want to argue that a competition measures* ***performance****, but a standardized test might measure* ***intelligence****.*

WHAT DO YOU THINK? Moderator

June 8, 2000

HOW TO MEASURE INTELLIGENCE?

A MEMBER OF THE ADVISORY BOARD MARVAN JABRI SAYS:

1. *A list of tasks/conditions can be defined.*
2. *Learning (on-line or off-line) can be included.*
3. *The generalisation capability is obviously critical.*
4. *The resources utilised are important.*
5. *Speed of learning (lapsed time of a trial and number of trials) would be important.*

In every community there are some benchmarks. Maybe the workshop can come up with a list of benchmarks that try to cater for various levels or dimensions of VI.

The difference between intelligent systems and autonomous systems is very vague. In some sense AI is like shooting on a moving target, what systems can aspire at doing today could be simple in the future. So ideally one would have a spectrum of tasks with various scales of difficulties. In other words something like a MIQ test.

WHAT DO YOU THINK ABOUT THESE STATEMENTS?

Moderator

June 13, 2000

ON THE UNIVERSALITY OF MECHANISMS OF INTELLIGENCE

By Thomas Whalen

I doubt that there are different kinds of intelligence. I know that multiple intelligences is a faith (very similar to the polytheism of ancient people).

It is easy to declare any manifestation of perceptual and cognitive activity to be an intelligence.

It is more difficult to find what do they have in common, maybe, absolutely the same.

We all can easily demonstrate that all known phenomena of intellect and intelligence are linked with a limited number of special computational algorithms including

--combining N-tuples

--searching

--focusing attention

--grouping (which includes "combining tuples")

--evaluating and ranking the results of grouping.

There are many humans who are superb at carrying out these five activities in one area but only average or occasionally even below average in applying the same activities in other areas. By 'areas" I mean things like math, music, human relations, mechanics, visual images, etcetera.

It may well be that to be considered "intelligent" in any field one has to be good at every one of these five within the context of that field, which would make the list a good universal definition of intelligence. But I suspect that the details of creating a constructed system that's good at these five things in one area will not be "plug and play" compatible with the details of doing so in another area.

It might be very instructive to look at just what is really measured by IQ tests and so on.

MODERATOR S COMMENTS

Tom Whalen agrees that probably the skill of intelligence consists of these five intertwined components: combining N-tuples, searching, focusing attention, grouping (which includes "combining tuples"), evaluating and ranking the results of grouping.

But he is worried that in many bright people these five components work in one context (domain) and do not work in another. Therefore, he asks: "Why people having this mechanism OK in one area cannot apply it within other areas"?

Probably, we all agree that these five activities constitute the body of the "mechanism of intelligence". Then, we are all surprised that it does not make a genius in literature to be a simultaneous genius in discrete math.

I think that we all are mistaken about it. The literature genius maybe is closer than we think to the discrete math genius (and vice versa).

Tom gave a hidden answer to this question:

-- because this mechanism works "within the context of that field".

The keyword here is "context":

[1. CONTEXT - The part of a text or statement that surrounds a particular word or passage and determines its meaning.

2. CONTEXT - The circumstances in which an event occurs; a setting.

(from American heritage Dictionary)].

The mechanism of intelligence is here but it works only in the language of a particular domain (thus, can read only the context submitted in this particular language).

How can we escape this predicament? Either we should translate the problem and the context into the language that the mechanism of intelligence understands, or from the very beginning, we should be able to operate in a "metalanguage" and translate contexts from all languages into the "metalanguage" (was proposed by E. Messina).

It seems reasonable to expect that the mechanism of intelligence working with excellence in a particular language 1 of a domain 1 can be easily retrained into working in a "metalanguage". Some humans have problems with this because they are enslaved by their prejudices about multiple intelligences. Machines are more advanced creatures: they to not have software prejudices unless one put them into an operating system.

Then, translation of a problem from languages 2, 3, etc. into the metalanguage -- is just a technicality. This is why we can expect that the box of "intelligence" can be context and even domain independent: it will work in metalanguage. Just put at the input a translator from the domain language into the metalanguage.

Do I expect that this is simple? Taken in account a thick bark of prejudices that the problem is (and we are) covered, probably not.

Can we help this process? Probably yes.

June 14, 2000

INTELLIGENCE AS A GOAL-BUILDER FOR THE CONTROL SYSTEM AND THE PARAMETRIC EVALUATION OF INTELLIGENCE

Cliff Joslyn (C. J.):

A. Meystel (A. M.) :

A. M.

1. If the goal is somehow obtained (constructed), then we should build a model of the system and apply Hamilton/Jacoby (H/J) and Euler/Lagrange (E/L). Actually, this is a reference to Calculus of Variations that allows to derive the laws of motion, dynamics, physics without any need to refer to experimental data. (In textbooks, you can find derivation of Newton's Laws, for example, F=ma by applying E/L equations to the cost-function assigned as the expression for energy).

C. J.

OK, like a generalized least action principle? It also sounds similar to Jaynes' derivation of thermodynamic laws from an entropic maximization constraint.

A. M.

Then, we can introduce planning of the future motion as finding the minimum cost motion trajectories by assigning ANY form for the COST. This means that cost is the primary factor, and since assigning COST depend on the goal, then the goal becomes the primary factor.

C. J.

OK, I think I follow you here.

A. M.

Of course, the goal presumes that there exists a source of the goal, and in many cases, this source exists as the carrier of INTELLIGENCE. For example, for a single level in the hierarchy of intelligence the adjacent lower resolution level (level "above") can be considered a source of the "goal".

C. J.

This is what emerges from this line of reasoning. A definition of the amount of intelligence in a system might involve a quantitative measure

of:

**) the amount of phenomena under control;*

**) the number of environmental distinctions measured by the system;*

**) the complexity of modalities of measurement and control;*

**) the complexity of the environmental variety available to the measurement and control of the system.*

These are all related to each other in complex ways, but the nub of it is there.

A. M.

I appreciate your compliance with the option of considering intelligence as a player. However, I am not sure that I can accept your FOUR SUGGESTED PARAMETERS that you consider a set of quantitative measures for intelligence. To me, these four factors are rather characteristics of a system that is associated with the use of intelligence.

C. J.

What's the difference? The "intelligence" of systems (and I do NOT advocate the use of this term in this context) is based on their manifesting a semiotic relation which has been selected by evolution or by designers, allowing the system to "choose" to act counter to physical law.

A. M.

Semiotic or non-semiotic - it does not matter if you do not define the PLAYER who WANTS and the PLAYER who PAYS THE COSTS (they might be the same). The term "semiotic" might obscure the essence of the situation that can reveal the phenomenon of intelligence. It looks like the essence is in an existence of a source of INTENTION.

C. J.

In attempting to reconcile your usages of terms with mine, I would say the following prerequisites necessary to find an intelligence in the control system. First, a goal state is necessary, provided from an external source, call it "a want" (an intention) provided by a player. The action of the control system is to maintain the system aware from its natural equilibrium, and this requires action and work. which can be identified as costs. And yes, the goal (ends) constrains the possible actions (means), and vice versa.

A. M.

We are interested in understanding the phenomenon of INTELLIGENCE and thus decided to model the system with the factor of INTELLIGENCE taken in account. Therefore, we should determine what plays the role of COSTS, what is the source of GOALS (WANTS), and naturally, I call this source a PLAYER.

I state this again and again because you (in your statement above) said: The "intelligence" of such systems is based on their manifesting a semiotic relation, and this statement mutes the emergence of a player with his/her WANTS<=>GOALS=>PLAYER.

C. J.

Apologies: the (perhaps implicit) presence of a goal state is, of course, necessary.

A. M.

I thought that this is pretty obvious that Powers/Marken do not want to introduce the concept of intelligence. They were not interested in this, they have other goals. The strive toward minimalism is not a new phenomenon. But there is a limit to our possibility to minimize the number of factors to be taken in account. When I refer to your state of "have overgrown" I refer to the FACT that initially we all are trying to cut the number of factors involved. One should notice that at some point of system's complexity it becomes detrimental.

C. J.

OK, I understand the admonition. My problem is always that until we can agree on these fundamentals, I have little faith that we can or should move onto the complexities.

A. M.

OK, I would agree with addressing an example of the Inanimate World with similar complexities.

C. J.

Then, consider the flipping coin. In this context Representation plays the role of the causal forces acting on the coin, and Will (Intention) is an abstraction of whatever ıt is which resolves the uncertainty as to whether heads or tails will turn up (call it Chance, or Chaos, or Statistical Physics).

A. M.

I would not start with this example. It is very complicated because we have two Wills here: the Will of the Man who is flipping the coin, and the Will of the Physical Law that we are not equip to compute since we do not know well enough the point that the force has been applied, the value of this force, the angle under which it was applied, the air resistance and so on.

C. J.

In the problem as set up, we ignore the will of the flipper. Thus in a sense the coin "wants" to come up heads or tails.

A.M.

Cliff, this is not so. The want of the coin is determined by the physical laws that are not well determined

C. J.

Since the chaotic flipping process is unpredictable, we cannot resolve their will into a physical explanation, but rather must resort to a statistical description. The complexity of those chaotic physical processes we simply bundle into the "will" of the coin: that which resolves the uncertainty, chance.

In your usage you can extend control, and thus intelligence, to any physical process. You are free to do this, but I find it unparsimonious, extending the term beyond any useful boundary. Instead, we need a principled way to distinguish control from other processes.

A. M.

Still, some terminological issues will remain blinking on the screen and demanding for future clarification.

C. J.

My conclusion is that on strictly denotational grounds, every control system can be seen as a semiotically closed system (NOT that "any closure is a semiotic system"), but that this is not the sense that H. Pattee intended.

A. M.

It does not matter as soon as it is true, constructive and useful. I would say even more (and this is what H. Pattee probably intended to say) that ANY INTELLIGENT SYSTEM IS SEMIOTICALLY A CLOSURE.

C. J.

Rather, Pattee is referring to the situation where the selection of the semiotic (coding) relations present in the system is itself a referent of that very semiotic system. This is thus not "simple" closure between a system and its environment, but between a system and its own construction or creation.

A. M.

Cliff, I have no qualms about it. It is not related to our discussion of intelligence. I would like to focus upon the set of issues that leads us to discovery, clarification, and better understanding of the phenomenon of intelligence. Obviously, the area of CONTROL SYSTEMS has its inner issues.

C. J.

I agree, I was only trying to distinguish between a literal sense of semiotic closure and H. Pattee's sense.

June 14, 2000

IS THERE AN ALGORITHMIC INVARIANCE WITHIN ALL KINDS OF INTELLIGENCE?

Walter Freeman has doubts about it and he responds to my discussion with T. Whalen in the following way:

Tom Whalen agrees that probably the skill of intelligence consists of these five intertwined components:

--combining N-tuples

--searching

--focusing attention

--grouping (which includes "combining tuples")

--evaluating and ranking the results of grouping.

W. F. These are pretty simple-minded, things MLPs can do.

YES, THIS LOOKS PRETTY SIMPLE-MINDED. BUT WE SHOULD NOT FORGET (I REPEATEDLY STRESSED IT) THAT THIS IS THE SET OF ELEMENTARY ALGORITHMS THAT FORM INTELLIGENCE OF A SINGLE LEVEL. AFTER GROUPING HAPPENS WE RECEIVE ANOTHER LEVEL OF RESOLUTION, A LEVEL WITH GENERALIZED OBJECTS. HERE THE SAME SET OF ALGORITHMS WORKS IN A SIMILAR WAY. AS A RESULT, WE RECEIVE ANOTHER LOWER LEVEL OF RESOLUTION, AND SO ON.

WALTER FREEMAN SUGGESTS TO TEST THE CONCEPT RELATED TO A SINGLE LEVEL. THIS CONCEPT THAT SEEMS TO BE TOO SIMPLISTIC SHOULD BE CAPABLE OF RESOLVING SOME SERIOUS EXAMPLES:

Try the following:

- Abstracting figures from undefined backgrounds
- Creating adaptive images of what to search for
- Prioritizing conflicting demands for mental workspace
- Generalizing and classifying items that are not linearly separable in n-space
- Translating between natural languages

I WOULD MEET THIS CHALLENGE WITH AN OPEN VISOR. LET US TRY TO SOLVE:

Test No. 1

Abstracting figures from undefined backgrounds

--I scan the image with a sliding window [SEARCHING] and store properties of the image [e.g. average intensity, color, etc.] at regularly selected coordinates.

--I hypothesize clusters based upon both properties similarity and adjacency [e.g. FOCUSING ATTENTION and COMBINING N-TUPLES]

--I promote clusters that I have discovered into a rank of objects [GROUPING]

--I am browsing my memory looking for similar objects [SEARCHING]

and so on.

Before I start browsing my images, I allow for some combinatorics upon created objects: the hypotheses of strings are considered together with their vicinities, and within the vicinity a local SEARCHING is executed (testing of combinations). This combinatorial freedom depends on the uncertainty of the results of clustering. When I perform browsing of my memory together with exploring combinatorial multiplicity of choices that comes from uncertainty.

(If the complexity of all this is too high, the problem distributes itself to other levels of resolution. This will reduce the complexity drastically).

What I have described in the previous two paragraphs is actually a solution of the second problem from W. Freeman's list:

Test No. 2

Creating adaptive images of what to search for

It would not be proper for me to go trough all examples of the list. But if one wants to do it, one will easily find that the solution for most of these problems can be represented by the five elementary algorithms that together are sufficient for modeling what some might call a "generic intelligence".

All cases of "gestalt" known from the literature allow for doing this.

I would suggest to all of you to make this and/or similar experiments. I am sure that if one have not resolved many similar problems earlier, it was only for the reason that one knew for sure that this is impossible. Sometimes, the expectation of futility of the possible effort is even more frightening than the complexity of problem.

It is really chilling to read something like: ...if you try to do this "you might wind up with is a collection of Turing Machines, that can talk to each other, but nobody else."

Sure, better even not to try...

In the meantime, if this collection will be a hierarchy of Turing Machines, the long term outlook might be very promising.

Moderator

June 14, 2000

WE CONTINUE TO DISCUSS THE ABILITY TO HAVE A CONTEXT-INDEPENDENT MODEL OF INTELLIGENCE

Paul Davis wrote:

(AND I WILL COMMENT AFTER EACH STATEMENT. A. M.)

P. D. Reactions to the set of five:

1. We probably need multiple levels and perspectives of intelligence's components. Quantum mechanics is beautiful, but it's not of much help to someone working at the levels of classical statistical mechanics, thermodynamics, engineering laws like Navier Stokes, or even cruder engineering scaling laws. The periodic table may be the essence of chemistry in some sense, but it doesn't take organic chemists very far. As the story goes in discussion of complex adaptive systems, different levels have their own laws.

NO DOUBT ABOUT IT. THE ELEMENTS OF THE GENERIC INTELLIGENCE (PAUL CALLS IT "THE SET OF FIVE" BUT IT MIGHT BE "SIX" OR "SEVEN") IS A SET THAT IS PRESUMED TO WORK AT A SINGLE LEVEL OF RESOLUTION. AS ONE CAN SEE, A PART OF ITS FUNCTIONING IS CREATION OF GROUPS, I. E. BUILDING UP A REPRESENTATION FOR THE NEXT LEVEL OF LOWER RESOLUTION.

SO, MULTIPLE LEVELS EMERGE AS A RESULT OF NORMAL FUNCTIONING IF THIS SET OF FIVE (OR SIX, OR SEVEN).

P. D. 2. Perhaps the set of five is a reasonable place to start discussion regarding ONE level/perspective. I suspect that it is incomplete, and I note the comments here of Walter Freeman. Beyond that, however, I wonder what empirical/theoretical basis exists for this or another set of underlying components or mechanisms. I would be very interested in a related discussion, because I'd learn a lot. But I don't believe that it would be nearly as useful as its proponents might hope (back to item 1, above).

PAUL IS CONFIDENT THAT THE PHENOMENON OF BEING MULTIRESOLUTIONAL IS MORE IMPORTANT THAN PROCESSES AT A SINGLE LEVEL. CERTAINLY! I AGREE WITH

YOU, PAUL. BUT THE MULTIRESOLUTIONAL SYSTEM OF REPRESENTATION EMERGES BECAUSE OF THIS "SET OF FIVE"!

P. D. 3. I would think that using Gardner's components of intelligence would not be a bad starting point from the other end, although others may have better suggestions.

AS YOU KNOW FROM MY PREVIOUS MESSAGES THE PHENOMENON OF MULTIPLE INTELLIGENCE IS EASILY TAKEN CARE OF BY INTRODUCING A TRANSLATION FROM THE DOMAIN OF APPLICATION INTO A NEUTRAL (META) LANGUAGE. IS THIS TRANSLATION IMPORTANT? OF COURSE! SHOULD WE DEVOTE ATTENTION TO THIS PHENOMENON? YES, OTHERWISE WE WON'T BE ABLE TO HANDLE IT.

P. D. 4 . While some may have the OPINION that multiple intelligences is a myth or an expression of "prejudice" (a rather inflammatory term), I have seen nothing in the e-mail to justify this opinion. The periodic table wasn't postulated or asserted; it was built up from empirical observations and minitheories.

PAUL, LET US GO TO AMERICAN HERITAGE DICTIONARY:

[Prejudice 1. a. An adverse judgment or opinion formed beforehand or without knowledge or examination of the facts. b. A preconceived preference or idea. 2. The act or state of holding unreasonable preconceived judgments or convictions.]

NO, I DON'T THINK THAT THE TERM "PREJUDICE" IS OR SHOULD BE TAKEN AS AN INFLAMMATORY ONE. THIS IS RATHER A TIMELY WARNING.

SPEAKING ABOUT PERIODIC TABLE: I WANT TO REMIND YOU, PAL, THAT THE PHLOGISTON THEORY WAS BUILT UP ALSO FROM EMPIRICAL OBSERVATIONS AND MINITHEORIES...

P. D. There is an extensive body of pyschological literature supporting--at that level of description--the notion of multiple intelligences (and the failure of the single G-factor hypothesis).

YES, BECAUSE THE IDEA OF INVARIANCE OF THE INTELLIGENCE (WITH AN INPUT TRANSLATOR) MIGHT BE DIFFICULT TO BEAR FOR MANY. INDEED, ONE MUST BE VERY RESPECTFUL OF THESE TONS OF SWEAT, BLOOD, AND TEARS SHED TO GAIN HIS/HER DOMAIN KNOWLEDGE. IT IS HARD EVEN TENTATIVELY TO ASSUME THAT ALL THIS IS JUST A TRANSLATOR WHILE REAL GENIUS IS A SYMBOLIC ALGORITHM! WOULD I VOLUNTARILY ADMIT THAT ALL HIDDEN TRICKS OF MY DOMAIN OF NUCLEAR PHYSICS ARE REALLY RESOLVED IN THE SAME WAY LIKE THE PROBLEMS OF CULINARY OR PLUMBING DOMAINS? NO WAY!

AGAIN, THIS IS THE ESSENCE OF THE HYPOTHESIS AT HAND:
1) AN INTELLIGENCE AT A LEVEL IS THIS SET [SEARCH*FOCUSING ATTENTION*GROUPING*SELECTION* (MAYBE SOMETHING ELSE)]
2) TOGETHER ALL OF THESE PRODUCE THE NEXT LOWER LEVEL OF RESOLUTION WHERE THE SAME ACTIVITIES ARE INITIATED
3) [AND SO ON]
4) TOGETHER, THE HIERARCHY OF THESE LEVELS IS EASILY COPING WITH NP-COMPLETE PROBLEMS
5) ALL MECHANISMS MENTIONED ABOVE CAN WORK IN THE SPECIFIC LANGUAGE OF A PARTICULAR DOMAIN AND EQUIP THEMSELVES WITH VARIOUS AND THE NEAT CORNER-CUTTING TRICKS APPROPRIATE FOR THE DOMAIN LANGUAGE.
6) AS FAR AS MACHINE INTELLIGENCE IS CONCERNED, ALL IT COULD BE DONE SYMBOLICALLY (IN A METALANGUAGE) IN THE SAME WAY IN ALL DOMAINS; JUST AT THE INPUT AND OUTPUT WE HAVE TO HAVE CORRESPONDING LANGUAGE(i)-->LANGUAGE(meta) AND LANGUAGE(meta)-->LANGUAGE(i) TRANSLATORS.

Moderator

June 14, 2000

FROM THE RESPONSES TO C.WEISBIN'S QUESTIONS

Thomas Whalen wrote:

I WOULD LIKE TO DISCUSS WITH ALL MEMBERS OF THE ADVISORY BOARD THE LIST OF QUESTIONS THAT I PROPOSED IN THE RESPONSE LETTER TO C. WEISBIN. LET ME KNOW WHAT DO YOU THINK, THIS IS VERY IMPORTANT.

This is the list of C. Weisbin's questions that the Workshop will try to answer:

1. What is the vector of intelligence (VI) that should be measured and possibly used as a metric for systems comparison?

*a) understanding instructions expressed in language convenient for the human giving them. *This is sometimes natural language, sometimes human-oriented technical language.)*

b) understanding goal specifications and working independently to achieve goals presented to it in a language and level of detail convenient to the human whose goals they are.

c) generating (sub)goals in a useful but surprising way so as to improve the well-being of the humans using the system.

2. Should VI be measured in addition, or instead of measuring the vector of performance (VP) determined by the regular specifications?

I think it comes to the same thing, just with a different emphasis

3. If two systems have the same VP, what is implied by the difference in their VI values? Can this difference be represented in $ units?

If VP does not include cost, then a more intelligent system would sometimes be more costly, sometimes cheaper. If VP includes cost, benefit, and risk, including all externalizes, then nothing else is economically interesting.

Example: it might be possible to someday build two Chinese rooms, one that "really understands" Chinese and the other which just follows stimulus--response rules. If so, the intelligent one will probably be cheaper to produce.

4. Is it possible (and meaningful) to have different VI measures:
a) goal-invariant, b) resource-invariant, c) time-invariant?

I don't understand the question.

5. What should be recommended as a test of VI and how to normalize VP so that comparison be performed at the same normalized value of VP.

While I don't think that VP and VI are identical, I don't see a sharp enough distinction to be able to "normalize." If a human has VI>VP we attribute it to poor motivation or else to a specific disability. If a human has VP>VI we attribute it to a fault in testing or to extraordinary motivation.

June 14, 2000

Kirstie Bellman responds:

Those Weisbin questions are a reasonable start. It will be interesting t see how quickly discussions emerge on the behavioral correlates of "understanding" within different environments or artificial ecosystems. (Kirstie Bellman)

June 15, 2000

ARE THE CONTEXT AND DOMAIN INDEPENDENT MODELS OF INTELLIGENCE POSSIBLE?

W. F. What I want to say is that intelligence is to be found in the capacity for defining objects, which requires action by the agent (robot, animal, human) in respect to goals that the objects are to make achievable.

A. M. You admit that this is something we can understand, model, and simulate with the help of computer. To make it clear you refer to the fact that:

W. F. This is straight-forward theory psychology from the pragmatist and gestalt schools, and it has been incorporated by a number of avant-garde roboticists.

A. M. And, yes, you admit that this straight-forward theories can be fully understood and even simulated with the help of computer: this is what is actually available

W. F. defining of 'objects' that are to be measured as 'n-tuples', sought, attended, grouped, and evaluated precedes these operations.

A. M. However, you firmly believe that DEFINING OBJECT is what we still cannot fully understand, and this is why computationally it cannot be done IN ALL CASES:

W. F. Once the objects are defined, Turing Machines will do fine, but Turing Machines can't do that . [defining the objects. A. M.].[I would add "in all cases" A. M.]

A. M. Yes, we have a problem with defining the objects if this is linked with our "wants". You finger exactly in this direction:

W. F. My only contribution is to show by modeling brain dynamics that biological brains have this capacity,

[defining the objects. A. M.] *and its exercise is well described by the theory of intentionality.*

A. M. Yes, but in numerous domains, we start implementing "intelligent systems" that are solving more simple problems of defining the objects. In many cases this operation is within our reach, and we perform it successfully. The efforts continue, sophistication grows. Then, I hope, that modeling of "intentionality" will be in our reach soon, too.

All of this was just an introduction to the expression of your big doubts concerning the concept that "intelligence" in various domains might be modeled by the same computational structure

W. F. I don't really object to proposing a common feature of 'generic intelligence', which may be in a class with other ideals such as truth, beauty and justice,

A. M. Walter, the only thing that I propose is to have a multiresolutional model of knowledge representation that will have at each level of resolution a model-set [searching*grouping*focusing attention*... ...*evaluation*selection] that will do a definition of objects at this level from the objects of the higher resolution defined at the level beneath.

It is my conviction that this system can work both in the domain language and in metalanguage. In the latter case, it can be considered a context-independent algorithm (model) of intelligence.

I am far from a desire to talk about spiritual and other hot air producing issues that are, as you are saying, "in a class with other ideals". Some participants of the discussion, called this context-independent intelligence: "generic intelligence".

I have no objections against any relevant term. For me, the essence of this is the most important issue.

In conclusion, you said:

W. F. but I doubt that it could support judgments more compelling than to say "system A is smarter than system B". To make it stick, you have to say what each can do, at what cost.

A. M. We will be able to say: "system A is smarter than system B" ? Not bad! We "have to say what each can do, at what cost" -- no doubt about it!
Moderator

June 16, 2000

TURING TEST, SUCCESS VS. LUCK, SUPERVISION AND AUTONOMY

Dan Repperger wrote:

Some comments after reading the white paper:

(1) Your efforts to quantify intelligence, especially from the perspective of a machine present a difficult problem. Your example of a Chinese room negates the Turing test as a possible definition of machine intelligence.

A. M. DON'T YOU THINK THAT IT IS A RIGHT TIME TO STOP JUDGING INTELLIGENCE
BY A SIMPLE SKILL TO PRETEND BEING "INTELLIGENT" ? (A. M.)

D. R. (2) The definition of J. S. Albus seems comprehensive enough and you transfer the responsibility to defining success, rather than being due to pure luck.

A. M. I HOPE THAT ONE COMPONENT OF THE VECTOR OF SUCCESS OF OUR MEETING WILL BE A CONSENSUS ON MEASURING THE SUCCESS OF IS FUNCTIONING

D. R. (3) Your vector of intelligence on page 5, I thought, would have a component of the speed at which it accesses information. It did not have this component but you address it later on.

A. M. YOU ARE RIGHT: THE SPEED OF ACCESSING INFORMATION IS A MAJOR ISSUE

D. R. (4) I agree that supervisory control and defining autonomy in subordinate systems is a key problem to be addressed in the next 10 years or so. The Air Force is very interested in this problem in the design of unmanned air vehicles.

A. M. SUPERVISORY CONTROL-->A DEGREE OF AUTONOMY-->AUTONOMY THAT MAXIMIZES EFFICIENCY -- PROBABLY, THIS WILL BE OUR PROGRESSION IN TIME.

Moderator

June 16, 2000

A RESPONSE TO THE DRAFT OF THE WHITE PAPER: FROM COMPUTING WITH WORDS → TO CHINESE ROOM IN REAL CHINA

I. B. Turksen wrote:

B. T. I have read your white paper with interest and enthusiasm. I agree with you that metrics of intelligence need to be developed. However, I would like to suggest that such metrics should be developed not just with "Computing with Numbers" paradigm in your reference to Lord Calvin, but as a synthesis of "Computing with Numbers" and "Computing with Words" paradigm of Lotfi Zadeh.

A. M. CERTAINLY, IT WOULD BE SUPERFICIAL TO UNDERSTAND THE NEED IN METRICS AS THE NEED IN A SOLELY QUANTITATIVE FORM, OR A FORMULA. ULTIMATELY, THE NEED IN A METRIC IS DETERMINED BY THE NEED TO COMPARE ALTERNATIVES, IN OUR CASE, TO COMPARE INTELLIGENT SYSTEMS: "WHICH ONE IS MORE INTELLIGENT," OR "WHICH ONE IS PREFERABLE FOR OUR NEEDS."

IF THIS PREFERENCE CAN BE FOUND WITHOUT NUMBERS AND RANKING CAN BE DONE AS A RESULT OF SOME LOGICAL INFERENCE -- SO BE IT!

B. T. Which you indirectly say but not clarify as one should. For example, you write: ...by living creatures, and especially by humans: ability to work under a hierarchy of goals, ability to perceive the external world and organize objects, actions and situations,... (quoted from the first page in the last paragraph). Note that humans at least do this with the use of their natural languages. Thus my point about the Computing with Words of Lotfi. Note that he recently began to talk about Computing with Perceptions .

A. M. ...WHICH REMINDS US THAT DETERMINING PREFERENCES MIGHT BE DONE NOT ONLY WITHOUT NUMBERS, BUT EVEN WITHOUT EXPLICIT LOGICAL INFERENCE: "AH! I LOVE THIS LANDSCAPE (OR THIS BEAUTIFUL FACE, OR THIS POWERFUL PAINTING). WELL, IN ALL THESE EXAMPLES THERE IS SOME INTERPRETATION OF PREFERENCE, AND THIS INTERPRETATION MIGHT BE DONE ON A PRE-LOGICAL LEVEL (IF IT EXISTS).

B. T. Let me put it in a different way. Recently I participated in a teleconferencing with some Italian Colleagues. They have developed an artificial Nose . They want to use their electro-mechanical device with novel sensor which provide lots of information. They have used principal component analysis, neural networks, etc. all numerical based analysis. But they are aware that they cant represent humans' ability, e.g., sense of smell, to detect variations in food, e.g. , cheese, and drinks, e.g., wine. These are however expressed with linguistic variable that humans use. Clearly fuzzy set and logic approach is a preliminary but effective way to begin and conceptualize such complex metrics of intelligence.

A. M. THIS IS ONE MORE ARGUMENT IN FAVOR OF NON NUMERICAL EVALUATION OF THE DEGREE OF INTELLIGENCE

B. T. In page 6, item (g) of the White Paper, you talk about (CIRCLE) why not also include (ellipse) and other more complex shapes?

A. M. YES, IN EVALUATION OF THE UNCERTAINTY FOR EACH COMPONENT OF THE VECTOR OF INTELLIGENCE ANY CONFIGURATION OF THE UNCERTAINTY ZONE CAN BE EXPECTED. I AM TALKING ABOUT "CIRCLE" BECAUSE IT SHOULD MEAN EQUALLY LARGE UNCERTAINTY FOR EACH COMPONENT OF THE VECTOR OF INTELLIGENCE.

B. T. In page 7 of the White Paper, you talk about "gestalt" concept. Is there a relationship between gestalt representation and its word representation and the potential semiotic representation and its interpretation?

A. M. WHEN WE ARE TALKING ABOUT "GESTALT" TODAY, WE ALL AGREE THAT THIS IS THE TERM THAT HAS BEEN INTRODUCED TO ACCOUNT FOR RECOGNIZING ENTITY FROM THE MULTIPLICITY OF ITS SEEMINGLY UNORGANIZED COMPONENTS. IT SEEMS REASONABLE TO EXPECT LINGUISTIC GESTALT SIMILAR TO THE GESTALT IN VISUAL PERCEPTION, AND GESTALT WORKING IN ANY SYSTEM OF SYMBOLIC REPRESENTATION, I. E. SEMIOTIC GESTALT.

B. T. In page 8 of the White Paper, J Searle and Chinese room experiment are mentioned. Let me tell you my personal experience in 1982 in Taipei Taiwan. A friend and I tried to locate a Bank with a map with Chinese characters. We were able to locate the Bank by matching the street labels on the map to the street labels on the street name plates. Even though we didn't know what the street names meant or how they were pronounced, we were able to find the Bank. Hence mission was accomplished and the goal was achieved without knowing the Chinese language.

A. M. THIS IS A FASCINATING STORY! BUT IT DOES NOT SAY ANYTHING GOOD ABOUT THE NOTORIOUS TURING TEST. INDEED, YOU DEMONSTRATED MULTIDIMENSIONAL, MULTIFUNCTIONAL INTELLIGENCE. FIRST, YOU WERE CAPABLE OF PUTTING IN CORRESPONDENCE THE NOISY 3D-REALITY OF THE CITY WITH THE NOT VERYCONGRUENT SYMBOLICS OF THE MAP. THEN, YOU DEMONSTRATED INTELLIGENCE BY UNDERSTANDING THAT THE CAPTIONS ON THE STREETS SHOWN IN THE MAP ARE THE SAME AS CAPTIONS ON THE STREET POSTS IN THE INTERSECTIONS. I APPRECIATE THE FACT THAT ALL OF THIS WAS IN A HIEROGLYPHIC SIGNS AND FINDING SIMILARITY BETWEEN HIEROGLYPHS IN DIFFERENT SOURCES REQUIRES INTELLIGENT OF SIMPLE SEARCH FOR SIMILARITY.

YES, YOU'VE DEMONSTRATED YOUR INTELLIGENCE!!!

BUT THE MAN SITTING IN THE CHINESE ROOM AND COMPARING WRITTEN HIEROGLYPHS BY SIMILARITY DEMONSTRATES ONLY A LITTLE BIT OF IT (AT LEAST HE UNDERSTOOD THE ALGORITHMS OF COMPARISON OF SIGNS AND SEARCHING IN A TABLE).

Moderator

June 16, 2000

DEFINITION OF INTELLIGENCE AND SEPARABILITY HYPOTHESIS

by B. Chandrasekaran

Consciousness is usually treated as an intrinsic property. I experience my being, but I don't experience your being. However, I usually hypothesize that you have the same property, consciousness. I'll never know for sure about you, just as you'll never know about me for sure.

Except for this theoretical caveat, we pretty much attribute to each other the property of consciousness and get on with our lives.

On the other hand, the term "intelligence" has both an extrinsic and an intrinsic connotation, depending on context. I watch an agent's behavior, and based on certain characteristics of the behavior, I may conclude that the agent's behavior is intelligent. In this sense, it is an extrinsic characterization. On the other hand, sometimes the term has an intrinsic connotation, that of having a mind, an entity that experiences being, experiences having thoughts, and so on. Thus, calling a thermostat intelligent is OK as long as it is intended as an extrinsic characterization (and as long as you agree with the criteria that were used for judging the presence of intelligence in thermostats).

Claiming intelligence in the intrinsic sense for them is much more problematic. In AI and cognitive science, people often slide from one sense of the term to the other without being aware that they are doing so. That is because, until very recently, there was no reason to separate the extrinsic and intrinsic senses of the term. The only entities that we called intelligent -- biological agents of various sorts, including humans -- showed intelligence extrinsically, and we were reasonably confident of attributing to them intelligence in the intrinsic sense.

*But technology has made it necessary to separate the two senses. It now seems theoretically possible to conceive of entities that *behave* intelligently -- have intelligence in the extrinsic sense. But it is much harder to be certain about their having intelligence in the intrinsic sense. We are missing the full panoply of the evidential basis that allowed us to abduce intrinsic intelligence from evidence of extrinsic intelligence in biological agents.*

The situation is not unique to the term "intelligence." There are other biologically based concepts that seemed pretty clear until recently, but now suddenly seem problematic. Consider the concept, "mother." One normally thinks that whether A is a mother of B is a matter of fact, not point of view. However, consider the case where woman A's fertilized egg is implanted in woman B's womb, and the infant that is born is immediately given to woman C, who adopts and raises the child.

*There is no self-evident answer to the question, "Who is the *real* mother of the child?". That is because contributing the egg, carrying the fetus in the womb and raising the newborn for several years are all typically done by one woman, and thus we normally do not separate these three properties associated with the concept of "mother." Depending on the purpose behind the question, however, we can answer the question. Thus, if the question is asked from the viewpoint of finding a donor for kidney, woman A is the mother. From the viewpoint of finding a woman who can suckle the infant, woman B is the mother. From the viewpoint of finding someone to solace the child when crying, woman C is the mother.*

I think that similarly, because of its natural orgin, at least two properties, perhaps more, come packed in one word "intelligence." If we don't recognize this and argue about what "really" is intelligence, and whether the thermostat is "really" intelligent, we will be like the people who argue about who "really" is the mother of the child in my story above.

It is possible to argue that the criteria used by Albus to attribute intelligence in the extrinsic sense to thermostats were too weak. Someone making this argument would hold that a meaningful characterization of extrinsic intelligence would seek to capture a much larger range of adaptation and behaviors than thermostats possess. Such an argument would identify higher mammals perhaps as a reasonable place to start, if not just focus on humans. But this is not a debate that has a clear correct answer either. One can choose one characterization as more interesting, more productive, and so on, but not as the one that is truly correct.

And, carrying this argument further, one might claim that the more complex forms of extrinsic intelligence can only be generated by systems that also have intelligence in the intrinsic sense. One way to interpret Penrose is that he is saying that the extrinsically intelligent behavior of a top-flight mathematician is not possible without certain essential characteristics of intrinsic intelligence. According to him, the mathematician directly "experiences" the truth of certain mathematical propositions. This capacity of intrinsic intelligence is essential for his behavior of finding a proof of the theorem.

While it may turn out to be true as a matter of empirical fact that we will only solve the problem of making artifacts that have a significant extrinsic intelligence only by making them have intelligence in the intrinsic sense, the latter is not logically a prerequisite for the former. At least, no one has shown it is. I have proposed what I call the "Separability Hypothesis" as a good working hypothesis for AI, namely, that it is not necessary to solve the problem of consciousness or intelligence in the intrinsic sense, to produce artifacts that show intelligence in the extrinsic sense. Those who are curious, see:

http://www.cis.ohio-state.edu/~chandra/separability.pdf

Chandra

B. CHANDRASEKARAN, OHIO STATE UNIVERSITY

June 17, 2000

THE IMPORTANT POINTS OF W. FREEMAN'S MESSAGE

Dear Advisory Board Members,

All of you received W. Freeman's message. I would like to emphasize some of the further developments that his message triggers.

1. ABOUT THE CONSCIOUSNESS

W. Freeman wrote:

I agree with Chandrasekaran, that "consciousness" need not be considered as a goal in machine intelligence, nor for that matter in biological intelligence and intentionality.

A. M.:

In other words, we have an additional support for the view that consciousness is a "GUI" for monitoring functioning of the system and its Umwelt.

2. INTELLIGENCE WITH AND WITHOUT LEARNING

W. Freeman wrote:

[An] intelligent system learns through practice. This rules out ordinary thermostats.

A. M.:

Is decision-making without learning "intelligent"? We can agree and accept that we will call "decision making without learning" a lesser degree of intelligence than "decision making with learning." It is possible even to postulate that a system with learning is supposed to be better performance, reliability, and so on. Of course, a cockroach learns ***not*** within a single generation. It learns at a lower resolution, at a specie level.

W. Freeman continues:

A child learns to recognize a spoon when it sees one, because it has practiced eating with it. Similarly, a machine can learn to recognize an electrical connector, if it can practice plugging itself in to recharge its batteries. You might say that a smart machine knows what it is doing, and a dumb one does not, but that invokes "knowing" of knowledge (or information), which is irrelevant to the design.

A. M.:

Walter, your last sentence would not be controversial if you rephrase it like this:

... a smart machine learns what it is doing, and a dumb one does not, and that invokes "learning" of knowledge (or information), which is relevant to the design since the devices for learning should be designed.

3. USING CHAOS AS A TOOL OF RANDOMIZATION

Then, W. Freeman said:

The learning in biological systems depends on chaos for hypothesis formation. This process is more closely related to statistics than to logic,

A. M.:

Walter, the statistics does not exclude logic, neither logic is fully complete without the logic of statistics. The logic of class formation, the logic of cause-->effect derivations, the associated deductions, inductions, and abductions neither disappear not lose their strength.

Randomization for hypotheses formation is a legitimate tool of reducing the complexity of computations. Chaos is a tool of randomization to collect more or less persuasive statistics. If chaos is generated that does not help to randomize properly, the statistical results may happen to be deceptive even if they look meaningful.

Therefore, biological system are not unique in using the tools of randomization for complexity reduction. Yes, they learned about these fascinating tool before Neanderthals, and before Cro-Magnons, and even before Haken s Research Institute in Europe and Santa Fe in US started exploring these things.

But in the engineering, these methods are utilized without too much associating these tools with mechanisms of intelligence.

W. Freeman:

[As] Johnny von Neumann wrote, brains "lack the arithmetic and logical depth" that we expect in machines, so he concluded that whatever the language of the brain might be, if it has any, it is "not mathematics, or at least not what we consciously and explicitly call mathematics" (1958). In other words, brains don't have numbers, but they do have a "way" of functioning which is highly successful in certain domains.

A. M.:

All musings of great people sooner or later become interpretable. In this particular case, we should not overestimate this "number-versus-another way" dilemma. In the previous letter, I. Turksen commented on Computing With Words paradigm that opens room for any symbolic system to be a language of the brain. I am sure that you would not reject the hypothesis that the language of the brain is symbolic (proof: based on the definition of "language").

W. Freeman:

That "way" is simulated in my KIII model (Freeman 2000). I look on it as a "machine-in-embryo", but it can already do useful work, such as reliable pattern classifications, that no other existing system can do, at all. Of course, it is simulated in software using numbers, so it is 100-fold slower than the sensory system it models, but it is realizable in hardware that could do the tasks in real time. The use of stochastic chaos instead of deterministic dynamics is what I perceive to be missing in your suggested approach. Am I wrong?

A. M.:

Walter, from my previous comments you could already deduce that randomization (and "stochastic chaos" is just a useful tools of randomization) is a regular and legitimate computational measure of complexity reduction. We probably are not aggressively propagating the word Chaos, we limit ourselves with a milder term "randomization" but - we have plenty of examples.

Today, NIST uses randomization at all levels of resolution of the Autonomous Intelligent Vehicle control. My first use of randomization can be dated to 1982-1984. In the system of Computer Aided Conceptual Design of robotic structures, I used it for hypothesizing assemblies. If one don't use it, the amount of computations grows unbearably[1]. But, I would agree, that "randomization" as a term would lose the contest with the term "Chaos".

Moderator

June 17, 2000

PAUL DAVIS ON THE DISCUSSION WITH B. CHANDRASEKARAN

Paul responds to my discussion with B. Chandrasekaran. Please, take in account that some thoughts concerning consciousness belong originally to B. Chandrasekaran (the letter is attached in the end).

Paul Davis wrote:
The term "performance" is associated with externally observable actions such as the maintaining a temperature task.

A. M.

[The specification of each target variable like "maintaining a temperature task" is actually more complicated. The specification sounds rather like: "maintain the temperature in the room within a particular interval of temperatures [from t1 - to t2]"] and in a more realistic scenario:
"maintain the expectation of the temperature within a particular interval [from t1 - to t2] while maintain the variance [sigma-t] within a particular interval [from sigma-t1 to sigma-t2]."

Then, each variable will be supplemented by the list of constraints like:

> [as you maintain what I asked above, please, do not exceed
> the total number of "on-off switching operations" higher
> than N; do not exceed the total consumed power (energy per time)
> higher than Pk, and so on].

[1] Paper by A. Meystel and M. Thomas on this subject was published in Proceedings of the IEEE Conference on Robotics and Automation, Atlanta, GA, 1984, pp. 220-229.

Paul, you are talking about the temperature. In the room, we always have a non-stationary arbitrarily shaped field of temperature. Usually, the temperature sensor is a part of the thermostat. It measures the value of temperature in the vicinity of the thermostat, and the temperature in other parts of the room is just assumed.

I would presume that a thoughtful engineer will install 5-7 sensors in different locations in the room and will require from this single thermostat to provide for, say, an average temperature to be within some interval. Donald Trump or a governmental diligent lab might be willing to distribute several thermostats in the room so that the particular temperature field be provided.

Paul Davis continues:

when the white paper talks about intelligence, it includes things like memory, processing speed, etc. In one frame of mind, these "sound like" other measures of performance. "Boy, that hummer really performs: it's a gigaflop machine!"

A. M. This sounds like a measure of performance only if your real problem is not (or cannot be) specified properly. There are many such problems: they are UNDERSPECIFIED, and we are interested to have some measure of the system universality and/or smartness so that the unexpected factors would not caught us unprepared!

BUT WHAT SHOULD BE CONSIDERED AN INDICATOR OF SMARTNESS: THIS IS THE SUBJECT OF OUR INQUIRY!

P. D.

Thus, one might think that we are going down the path of saying that both performance and intelligence are about doing tasks (e.g., crunching a billion arithmetic operations).

A. M. I hope that all we have already understood that

a) in a well specified case this is plain WRONG

b) in the underspecified case to rely on some buzzwords from commercials (for laymen as well as for the scientists) is plain silly,

or as Paul is saying UNSATISFYING.

P. D. That is correct - unsatisfying. It would hardly help with the Chinese Room problem. Even adding items such as number of objects discerned, or number of levels of resolution used, still don't sound like intelligence. Hmm.

A. M. The greatness of the moment is in the fact that we have realized already that the characteristics of the system can and should be looked at carefully. In our white paper we have divided them into two groups:
VECTOR OF PERFORMANCE (that characterizes the output variables) and
VECTOR OF INTELLIGENCE (that characterizes the properties and features of the system of control)

Paul whose language is a little bit different prefers to talk about
EXTERNAL (equivalent to OUTPUT, or PERFORMANCE set of variables), and INTERNAL (equivalent to INTELLIGENCE set of variables).

Then, the picture looks for him more peaceful:

P. D. There are two parts to the solution, I think. First, we need to distinguish between internal and external. If a machine has Gigaflop capability, that is an internal capability related to potential intelligence. To be sure, we might measure that capabilitiy by having the machine do a task. However, it might also be possible to study the "anatomy" of a machine and infer that it has parallel processing capability; if so, that would be another way to infer internal capability--one that doesn't require having the machine "perform."

A. M. But, of course! To study the ANATOMY of our intelligence is probably the only way to judge upon the future functioning especially if we are uncertain about it!

P. D. If it were possible to read some of the machine's programming, then we might infer that it has the capability to "detect" objects of certain classes, or even to give names to patterns that it "discovers." We might not know how the machine would perform at tasks, because we might not understand the totality of the programming, but we could at least see potentialities.

A. M. Hurray! This is one of the best descriptions of the Vector of Intelligence ONE could ever dream. This is how COMPUTER VISION people are talking about their systems: in the terms of classes available for distinguishing and interpreting. This is how PLANNING/CONTROL people are talking about their systems: in the terms of classes of terrain they can handle, and classes of obstacles that they can efficiently avoid. In both cases: vision and planning/control the description of classes can be general enough, yet adequately presenting the properties of the future problems.

P. D. It seems to me that, so far, we are on good ground distinguishing between performance and components of intelligence, or, perhaps, enablers of intelligence VI. Moreover, it seems to me that none of this is mystical.

A. M. In other words, we are capable of specifying for all cases the relevant VP and VI.

Paul Davis goes on:

The other part to the solution requires doing something about the "emergent properties" business. Some aspects of intelligence seem to demand this. I don't think that we can claim to have tackled intelligence without at least building place holders in for capabilities such as world modeling, including world modeling that adds inferred features that were not already lurking in the machine's data base.

A. M. These are formidable observations! We, the people, all our life are doing one thing: (as Paul describes it) searching for "inferred features that were not already lurking" in our memory. Obviously, the contemporary audience has learned about the "emergent properties" first and only after this it has noticed that it actually infers new features.

[The next paragraph is more related to positions from
the B. Chandrasekaran's letter (see a couple of letters back)]

P. D. And what about emotion and its machine analogues if there are some? Perhaps Crick's work is relevant here (e.g., his book using vision as something we more nearly understand that may be related to consciousness). Perhaps it would be possible to determine whether machines have and adapt internal models of the external world by giving it certain tests. We wouldn't necessarily "see" the machine's model (unless it was a simple program that we inserted in the first place), but we might be able to have a strong basis for inferring the existence of a model. Going back to our furry friends for examples, some of us believe that certain of their actions go beyond something explainable by straightforward "behavior:" we would argue that the animals are "thinking," although we haven't a clue what their mental "picture" or reasoning is like in any detail. We could be wrong (and, certainly, some scientists are vociferous in insisting that other animals don't think), but our inferences have some basis.

[With this background, here are some P. Davis responses to the questions earlier presented to Advisory Board Members. See Attachment 1]

1. Yes, I think there is a difference between VI and VP. One has to do with internal capabilities; the other has to do with accomplishment of externally observable tasks. The first may be inferentially measured by having the machine do tasks.

2. VI should me measured in addition to VP.

3. If two machines have the same VP, it might be because we only had a meager set of tasks and, as a result, didn't make the distinctions we might have. It certainly seems unlikely to me that we shall soon be able to infer VI from VP.

4. Some aspects of VI (the enablers of intelligence, if not intelligence itself) might be goal invariant, such as processing speed, memory, etc. I'm not sure, however, what is meant by the several invariances.

5. This is a really tough question and I don't understand yet how to do it, except in some very simple respects. Getting at the existence and richness of internal models seems important here.

(the question was: 5. What should be recommended as a test of VI and how to normalize VP so that comparison be performed at the same normalized value of VP. A. M.)

6. I would think that we could construct broad problem spaces, measure performances that give us hints about intelligence components, etc., without focusing on any particular problem area. If we did, however, the result would not be context independent, but rather information about how intelligence varied with context! Unless, of course, we did some gross averaging. I am very skeptical about simple measures in this business.

7. I think that resources are relevant, but shouldn't be allowed to dominate.

[This is the end of P. Davis' commented message]

Attachment 1

This is the list of questions that the Workshop is to answer:

1. What is the vector of intelligence (VI) that should be measured and possibly used as a metric for systems comparison?

2. Should VI be measured in addition, or instead of measuring the vector of performance (VP) determined by the regular specifications?

3. If two systems have the same VP, what is implied by the difference in their VI values? Can this difference be represented in $ units?

4. Is it possible (and meaningful) to have different VI measures: a) goal-invariant, b) resource-invariant, c) time-invariant?

5. What should be recommended as a test of VI and how to normalize VP so that comparison be performed at the same normalized value of VP.

The subsequent supplementary questions are ingrained (directly, or indirectly) in the main five questions:

a) how to form VI for various architectures?

b) should the questions 1 through 5 be related to intelligent systems, or autonomous systems, or both?

c) what is the protocol of dealing with uncertainty when the uncertain metric is to be applied in the procedures of decision making? for example how the uncertainty of planning affects the cost of goal achievement?

d) what are the guidelines in constructing the world model and determining its scope in the variety of applications? how the scope of "world model" affects the sophistication of intelligent behavior?

e) how are the questions 1 through 5 related (and the answers applied to) the systems that are working under a hierarchy of goals.

f) should a competition between intelligent systems be considered a valid method of judging VI value?

June 17, 2000

LEARNING, GOALS, INTELLIGENCE: COMPARATIVE AND DEVELOPMENTAL PSYCHOLOGY

Thomas Whalen wrote:

I think another good way to get perspective on the questions we have been wrestling with is to look at biological systems other than fully developed humans. There has been some discussion of animals already, but I think it could be made more systematic.

All biological organisms manifest learning by their genome, and what the socio-biologists call the "goals" of the genome. Whether this is per se a manifestation of "intelligence" is a metaphysical question.

An insect's behavior shows only this kind of learning, like a thermostat's behavior manifests learning and intelligence but not learning and intelligence of the thermostat's own, just the learning and intelligence of the thermostat's designers.

Higher animals such as mice show learning of their own. The trained behavior of a simple neural net or even regression equation also manifests learning of the system's own, but the goals and intelligence of the designers.

Carnivores like dogs and cats, and even more so primates, seem to have goals of their own beyond the goals given by instinct. Do our current autonomous constructed systems have goals of their own in this sense?

Does a gorilla or chimpanzee, especially one who has learned to use language or at least a language-like systems, have intelligence of its own?

More to the point, what does the question mean? Is it the same question as when we ask it about a constructed system?

Coming from another direction, a newborn infant's brain is physically immature as well as having vast amounts of learning ahead of it. Very young babies already manifest rapid learning of their own, overshadowing billions of years of genomic learning. But a newborn baby does not manifest intelligence of his or her own, while a five year old certainly does. If we watch the emergence of intelligence in children and collectively introspect about when we want to use the word "intelligent" we amy learn things useful in answering the same questions about constructed systems.

As an example, does the intelligence of a little child emerge as a unified phenomenon or do some "kinds" of intelligence emerge before others? Does the "vector of Intelligence" emerge in concert, or one element at a time?

I hope to break loose some time to review current comparative and developmental psychology looking for slues we can carry over into understanding constructed systems.

Tom Whalen
July 3, 2000

Answer by the Moderator:

Tom,

It is my deep conviction that your questions, both legitimate and interesting, CAN be addressed and answered within the formalisms that we use in the multiresolutional nested hierarchical planning/ control systems, when we introduce learning phenomena.

(see URLs: http://www.ee.umd.edu/medlab/papers/Final/Final.html
which serves to enable a structure similar to
http://www.ee.umd.edu/medlab/papers/trep/trep.html
or ask me about published references)

It is true, presently we do not build intelligent systems that have instincts enabling them to get involved in efficient learning. Living creatures have an interesting distinction from intelligent constructed creatures. Our robots are concerned with survival of

themselves
their team
their master
an assigned object.

Living creatures are concerned in addition with survival of their species, and we even don't know how exactly this thing is implanted within their architectures. My robot will defend, or bring information to his own team, to me and to an assigned subject because this is what I have implanted explicitly into its architecture, or its knowledge base.

Some people are saying: We can implant the ability for evolutionary development of the specie of my robot. We just cannot wait a couple of billions years to see how it will develop. They are right. We will have a problem of funding if we propose a type of research that should be even 100 years short.

Clearly, analyzing goals implanted into genome within the constructed unmanned robots, seems to be impractical. It is much easier to manipulate with its software at the stage of design. Thus, I cannot wait until he develops a feature, or a goal: I must predict what I want and prescribe both the goal and the feature.

Saying this, I do not feel sadness. I feel joy.

Maybe, because I was constructed to be an engineer.

Moderator

ARCHITECTURES OF INFORMATION EMPLOYED BY INTELLIGENT SYSTEMS

C. Landauer responds to A. Wild s Questions

C. L. We have been studying exactly the kinds of questions that Andreas Wild suggested, and we have answers for some of them and approaches for others.

A.Wild's questions:

1- How could an information system be architected such that heterogeneous elements, like different types of computation (reasoning ?), may coexist, interact and add value to each other ? What would be the interfaces between sub-domains looking like?

2- Can such a system evolve by including domains that became relevant after the system was built, or by modifying or eliminating some of the domains implemented at its "birth" ?

3- Is there a way for a system to control interfaces among its own sub-systems, e.g. define new ones, eliminate or modify existing ones ? Can a system re-architect itself ?

4- Can this happen across hierarchical boundaries without generating unbearable chaos ?

5- Is a non-hierarchical, self-configuring, heterogeneous system at all possible ? If yes, are there any rules to follow, are there impossible situations to avoid, or, alternatively, anything goes, and the solutions will be selected by trial and error ?

C. L. These are our[2] answers / approaches / expectations / hopes:

1-We have been writing about Constructed Complex Systems for some time now, providing them with a Knowledge-Based infrastructure that supports exactly this kind of heterogeneity, and a further property called Computational Reflection, which means that the

[2] K. Bellman and C. Landauer

system has a complete model of its own behavior (internal and external), to some level of detail, so it can examine and change its own functions and behavior[3].

2-We have added domains that only became relevant after the fact, since the system can defer until run time its search for relevant resources to apply to a problem (including the problem of interpretation of the problem statements in a language not defined until later in the system lifetime) - the system has no privileged resources at all, so anything can be changed, on the fly — this is partly achieved by explicit and uniform separation of the problems posed during system operation and the resources used to address those problems (the Problem Posing Programming Paradigm), and flexible mappings from problems to resources (Knowledge-Based Polymorphism), that together lead to a new approach to Generic Programming[4].

3-We have an approach to the creation and management of internal interfaces in a system, based on a new knowledge representation structure we called a conceptual category , which separates the purpose of an interface from its appearence in code (we have found the term used much earlier for something different, but we intend to keep it for this data structure anyway, since it is the right term for what we are trying to model)[5]. We are even trying to arrange that a system can change its own basic symbol systems, since we have shown that it must, if it is to persist for an extended time in a complex environment[6].

4-We think so, but have not proven it

5-We think so and are trying to prove it - the key here is that not all of the computations can be in the application domain - many of them have to be in the ''organization of the computing system'' domain, that is, much more internal infrastructure needs to be available than

[3] How this works with autonomous computing systems is discussed in the following paper: C. Landauer, K. L. Bellman, ''Computational Embodiment: Constructing Autonomous Software Systems'', pp. 131-168 in Cybernetics and Systems: An International Journal, Volume 30, No. 2 (1999)

[4] This architecture is described in C. Landauer, K. L. Bellman, ''Problem Posing Interpretation of Programming Languages'', Paper etecc07 in Proceedings of HICSS'99: the 32nd Hawaii Conference on System Sciences, Track III: Emerging Technologies, Engineering Complex Computing Systems Mini-Track, 5-8 January 1999, Maui, Hawaii (1999); C. Landauer, K. L. Bellman, ''Generic Programming, Partial Evaluation, and a New Programming Paradigm'', ibid., Track III: Emerging Technologies, Software Process Improvement Mini-Trackibid; revised and extended as C. Landauer, K. L. Bellman, ''Generic Programming, Partial Evaluation, and a New Programming Paradigm'', Chapter 8, pp. 108-154 in G. McGuire (ed.), Software Process Improvement, Idea Group Publishing (1999)

[5] The approach is partly described in C. Landauer, ''Conceptual Categories as Knowledge Structures'', pp. 44-49 in A. M. Meystel (ed.), Proceedings of ISAS'97: The 1997 International Conference on Intelligent Systems and Semiotics: A Learning Perspective, 22-25 September 1997, NIST, Gaithersburg, Maryland (1997)

[6] The proof and discussion are in C. Landauer, K. L. Bellman, ``Situation Assessment via Computational Semiotics'', pp. 712-717 in Proceedings of ISAS'98: the 1998 International MultiDisciplinary Conference on Intelligent Systems and Semiotics, 14-17 September 1998, NIST, Gaithersburg, Maryland (1998)

in most systems, and it needs to be much more capable and knowledgable than in most systems (it is clear from AW'sdescription that this was at least part of the problem for the example)[7].

We think we are making progress on all of these fronts, though more slowly than we would like.

Christopher Landauer

July 4, 2000

LARRY REEKER SHARES HIS THOUGHTS:

1. We Measure In More Than Numbers Alone

Though numbers are very useful, other entities are always involved in measurement, in various ways. The most obvious entities that are involved are linguistic ones. Words are used to indicate the dimensions of the measurement (mass, time, etc.) and the units used (kg., lb., sec., etc.). But there are other linguistic means by which caveats on the measurements are made, and these may be both numerical and non-numerical.

Numbers were invented for purposes of measurement, either of size (cardinal numbers) or of sequence (ordinal numbers). They were invented because they can provide a succinct, precise means of expressing size or sequence, and that is why people like them and have faith in them. But that can be a disadvantage, and qualifications are therefore necessary. One type of qualification has to do with precision or possible error, and other numbers can be used to measure such a qualification.

Additionally, there are descriptions of the data on which measurements are made. In information storage and retrieval, Recall and Precision are measures made numerically, but the text corpus over which the measurements are taken is important in interpreting these numbers. Similarly, the terrain over which a robot's navigation abilities are measured has a lot to do with the relevance of the evaluation to particular applications.

[7] An overview of the approach can be found in C. Landauer, K. L. Bellman, ''Architectures for Embodied Intelligence'', pp. 215-220 in Proceedings of ANNIE'99: 1999 Artificial Neural Nets and Industrial Engineering, Special Track on Bizarre Systems, 7-10 November 1999, St. Louis, Missouri (1999)

Despite qualifications, there is often a tendency to misuse numbers just because they are so handy and seemingly clear, whereas the qualifications are tedious and boring, often a little like reading an insurance policy or a legal contract. Thus one can be tempted to think that there are "lies, damned lies, and measurements" (to rephrase a popular adage about statistics).

I guess all of that is pretty obvious. It is also obvious that there are in principle ways to measure that are not numerical, as long as they involve lattice relations. One can define such a relation on a vector of numbers, and thereby rank the vectors according to some criterion. One is just ordinary normalization, but one can use weightings, too. For a particular application, a system evaluation vector could be multiplied by a vector that characterizes the needs of the system.

2. Lord Kelvin (A Historical Digression)

I have always had some reservations about Kelvin's famous statement on measurement, since really numerical measurements are neither necessary or sufficient for a scientific theory. Even as he wrote or uttered his famous statement,

"I often say when you can measure what you are speaking about and express it in numbers, you know something about it, but when you cannot measure it, when you cannot express it in numbers, your knowledge of it is of a meagre and unsatisfactory kind...", he had to know that numbers are not a sine qua non for measurement.

He knew, of course, that line congruence was the equivalent of numerical comparison in Euclid's geometry, but not expressed in numbers, so that one could clearly find ways to express significant facts about significant domains without numbers. (Despite the breadth across which his brilliance was spread, he was first a mathematician, and the son of a mathematician.) His aphoristic claim that is sometimes cited, "To measure is to know", is probably a better statement of his real feelings, for that reason. But Kelvin knew how to say things that would be remembered (no wonder that he eventually became a dean!).

Kelvin also said "If you can not measure it, you can not improve it", which is certainly true in the wider sense that the Workshop on Performance Metrics in Intelligent Systems is about, even if we take as the measurement device contests, or even human judgments of gradation. It would be very difficult to define "improve" in a sense that is not circular, without expressing some measurement concept. "To enhance in value or quality" is a common definition, and value and quality refer to measures, even if subjective.

It is unlikely, however, that Kelvin would have accepted any measures that were overly subjective. He might well have asked for more than Turing's "imitation game", test if he had been alive later. If we look at the context of the statement above, we see some expression of qualifications that are also revealing:

"In physical science the first essential step in the direction of learning any subject is to find principles of numerical reckoning and practicable methods for measuring some quality connected with it. [Here follows, "I often say that", as cited above.] but you have scarcely in your thoughts advanced to the state of Science, whatever the matter may be."

Here too, I would guess (for the reasons mentioned above) that "numerical" means "mathematical" in a wider sense than just numbers.

I would argue that in the last part of the quotation, Kelvin is wrong, but that he was making a valid point within the prevailing Anglo-American view of science in his time. Though Kelvin had been born almost 200 years after the death of Francis Bacon, he still lived in an era where the predominant view of science was due to Bacon, with contributions of Hume and other philosophers. Charles Darwin, working in roughly the same time frame that Kelvin was growing up, tells us that he (Darwin) "worked on true Baconian principles, and without any theory collected facts on a wholesale scale"

This collection of facts "on a wholesale scale" often lacked any more than subjective observation, and Kelvin's notion that measurement was important was potentially, in that framework, an important way of improving the inference of causal laws and disentangling phenomena that are only superficially related. But (to return to his contemporary Darwin) Lord Kelvin should certainly have recognized the contribution of the leap from collections (no matter how large) of phenomena to models (of which Darwin's "theory of evolution" was a model that advanced science materially). Had he done so, he would not have declared that such contributions (others of which he himself had made, as well as Darwin and others of the period) "scarcely advanced to the state of Science..."

3. Back to Measurement (After A Short Philosophical Digression)

Today, our view of scientific theory has changed from that held in the 19th Century. The bare-bones version of a scientific theory is that it consists of a model composed of abstract theoretical constructs and a calculus that manipulates these constructs in a way that can account for observations and accurately predict the value of experiments. The theoretical constructs have a relation with observed entities, properties and processes that may be quite abstract, not necessarily readily available to human senses, but following directly from calculations based on the theory. There are a number of principles applied to a model that give us increased confidence in the theory, but the one most relevant here is that we can measure the observed entities to confirm the predictions of the theories.

It is relevant to observe that the "calculus" mentioned above is used in the dictionary sense "a method of computation or calculation in a special notation (as of logic or symbolic logic)". That means that it may be numerical or non-numerical. In fact, as Herb Simon and Allen Newell pointed out so strongly almost a half century ago, that calculus might be expressed in the notation of a computer program, the better to speed its manipulation of the theoretical constructs.

With respect to the field of Artificial Intelligence, the point has sometimes been made that the value of individual research results is difficult to confirm, and this has been used to cast aspersions on the entire AI enterprise. It is not uncommon for this criticism to be made, at least implicitly, by people whose own knowledge of AI is “of a meagre sort”, and then one suspects that it is motivated by feelings of the sort that Herb Simon was describing when he wrote:

“I continue to marvel at the fact that, after 45 years, the naysayers can still be taken seriously, when they deny that computers (sometimes) think, or place that happy possibility in the distant future. I am afraid that at the outset of our adventure I greatly underestimated the emotional need many members of our species have to believe in its uniqueness… Patience! All that will pass. (Herbert Simon, E-mail, 26 Jun 1999).

On the other hand, there are well-informed AI critics whose views often reflect the fact that systems have been described qualitatively in ways that cannot be backed up by objective evaluation. There have indeed been system developers with simplistic ideas about scientific theory, suggesting that their computational models were theories of actual human cognition merely on the basis of surface resemblance. These suggestions have also been of concern to the majority of scientists in the AI and Cognitive Science Communities, who have been in search of solid theoretical concepts to underpin the field. The issue here is not necessarily the fact that the measures are qualitative, however. It is the fact that they are not meaningful as part of a scientific theory. They are therefore vague in a way that Kelvin would not have admired; but more importantly, they cannot as easily lead to improvements, as Kelvin so aptly observed.

4. Some Implications for Intelligent Systems Metrics

Perhaps (and Lord Kelvin would like this idea, I hope) our emphasis on finding metrics can solidify the theoretical constructs of the field, as well as providing a means of measuring progress.

The key to doing this is not to think of evaluation only as measurement of some benchmarks or physical parameters (which I will call "behaviors") that are manifested in the operation of the systems being evaluated. We need to think in terms of the inner workings of the systems and how the parameters within them relate to the measured externally manifested behaviors.

Consider the measures of Recall and Precision as an example. Given a particular text corpus, one can consider various weighting schemes, use of a thesaurus, use of grammatical parsing that seeks to label the corpus as to parts of speech, etc., within a system and see how these items (I realize they are more resources than theoretical constructs) relate to precision and recall in the context of a particular corpus, or of a corpus with particular characteristics (these might be theoretical constructs). I believe this sort of thing has been done, but it is not a field that I have followed recently.

It may be more interesting in the Workshop to consider component systems for things like reinforcement learning (RL). There are a number of different techniques within the RL, all of which have many possible applications. The concepts include the states chosen, the reinforcement function, and the policy. The area is becoming quite sophisticated, and there are known facts about the relation of these to outcomes in particular cases.

Suppose that a reinforcement learning system constitutes a part of the intelligence of an intelligent system. There should be some way of predicting how that system would do upon encountering problems of a certain nature. By knowing how it chooses the concepts in its system and how they react on problems of that type, one can provide a partial evaluation of how effective the learning system would be. By obtaining such figures for all such subsystems, one could relate them to the performance of the full intelligent system.

I am working on a general framework of this sort, and hope to discuss it further at the Workshop. I hope that we will eventually be able use such a framework systematically relate measures, whether numerical, non-numerical, or a combination thereof, at all levels of the system, from internal capabilities to external performance, to, as Lord Kelvin might say, "advance to the state of Science".

APPENDIX B
WORKSHOP SCHEDULE

Performance Metrics for Intelligent Systems

Workshop Schedule

General Chair — Elena Messina
Program Chair — Alex Meystel

August 14 - 16

The Workshop opens in Lecture Room A, Bldg. 101

Afternoon Plenary Lecture will be conducted at Green Auditorium

N I S T
Gaithersburg, MD
2000

Advisory Board

- T. Shih, Tamkang University, Taiwan
- R. Simmons, Carnegie-Mellon, USA
- M. Swinson, DARPA, USA
- M. Tilden, Los Alamos National Lab., USA
- I. B. Turksen, University of Toronto, Canada
- C. Weisbin, NASA, USA
- T. Whalen, Georgia State University, USA
- A. Wild, Motorola, USA
- V. Winter, SANDIA, USA
- J. Xiao, NSF, USA
- R. Yager, Iona College, USA
- A. Yavnai, RAFAEL, Israel
- Y. Ye, IBM T. J. Watson Research Center, USA
- B. Zeigler, University of Arizona, USA
- L. Zadeh, University of California at Berkeley, USA

The Schedule of a Session

- Each Session is allotted 2 hours.
- It is expected that a speaker will use a slot of 25 minutes for his/her presentation (20 minutes) and answering questions (5 minutes).
- The remainder of time (20 minutes) should be used for a general discussion and combining the *Final Recommendations* of the Session.
- The Final Recommendations of all sessions will be integrated into *Final Recommendations of the Workshop.*
- The results of Each Day are discussed at the evening Plenary Discussion

1st Day, Monday, August the 14th

First Day starts at 8.30 AM with introductory presentation:

J. Evans, E. Messina, Performance Metrics for Intelligent Systems

PLENARY LECTURE — 9 AM — 10 AM

H. Szu, Machine IQ with Stable Cybernetic Learning With and Without a Teacher

Coffee Break: 10 AM-10.30 AM

Sessions: 10.30 AM — 12.30 PM

I Day, morning A: Features of Industrial Intelligent Systems,
Co-Chairs: M. Cotsaftis, W. H. VerDuin

- M. W. Bailey, W. H. VerDuin, FIPER: An Intelligent System for the Optimal Design of Highly Engineered Products
- S. A. Wallace, J. E. Laird, K. J. Coulter, Examining the Resource Requirements of Artificial Intelligence Architectures
- C. Peterson, A Metric for Monitoring and Retaining Flight Software performance
- M. Cotsaftis, On Definition of Task Oriented System Intelligence

I Day, Morning B: Features of Living Intelligent Systems
Co-Chairs: K. Bellman, C. Joslyn

- K. Bellman, Understanding and Its Behavioral Correlates
- C. Joslyn, Toward Measures of Intelligence Based On Semiotic Control
- H. Sarjoughian, B. Zeigler, Model-based Design and Measurement of Intelligence
- T. Chmielewski, P. Kalata, Biometric Techniques: The Fundamentals of Evaluation

I Day, Morning C: Special Issues of Evaluating Intelligence
Co-Chairs: R. Sanz, A. Wild

- R. Sanz, I. Lopez, Minds, MIPs, and Structural Feedback
- A. Wild, Using the Metaphor of Intelligence
- R. Garner, R. N. Bishop, Applied Applications for Mimetic Synthesis: The AAMS Project Summary
- H. M. Hubey, General Scientific Premises of Measuring Complex Phenomena

Lunch 12.30 PM — 2 PM

PLENARY LECTURE — 2 PM-3 PM

G. Saridis, Definition and Measurement of Machine Intelligence

Coffee Break: 3 PM-3.15 PM

Sessions: 3.15 PM — 5.15 PM

I Day, Afternoon A: Metrics and Comparison of Alternatives: General Issues

Co-Chairs: L. Pouchard, W. C. Stirling

- L. Pouchard, Metrics for Intelligence: the Perspective from Software Agents
- J. Spall, et al, Towards an Objective Comparison of Stochastic Optimization Approaches
- W. C. Stirling, R. L. Frost, Intelligence with Attitude
- S. Lee, W.-C. Bang, and Z. Z. Bien, Measure of System Intelligence: An Engineering Perspective

I Day, Afternoon B: Metrics and Comparison of Alternatives: Case Studies

Co-Chairs: R. Finkelstein, E. Grant

- E. Grant, G. Lee, Properties of Learning Knowledge Based Controllers
- V. Grishin, A. Meystel, Using Visualisation for Measuring Intelligence of Constructed Systems
- R. Finkelstein, A Method for Evaluating the IQ of Intelligent Systems
- L. Polyakov, In Defense of the Additive Form for Evaluating Vectors

Plenary Discussion- 5.15 PM — 6.15 PM

Panel: K. Bellman, M. Cotsaftis, R. Finkelstein, E. Grant, C. Joslyn, C. Peterson, L. Pouchard, W. C. Stirling, A. Wild

8 PM — Meeting of the Advisory Board (at "Holiday Inn")

2nd Day, Tuesday, August the 15th

PLENARY LECTURE — 9 AM-10 AM

J. Albus, Features of Intelligence Required in Unmanned Autonomous Vehicles

Coffee Break: 10 AM-10.30 AM

Sessions: 10.30 AM — 12.30 PM

II Day, Morning A: Measuring performance
Co-Chairs: A. Sanderson, T. Samad

- T. Samad, Technologies for Engineering Autonomy and Intelligence
- A. Sanderson, Minimal Representation Size Metrics for Intelligent Robotic Systems
- J. Zhang, A Formal Method to the Performance Metrics for Engineering Systems
- R. Yager, A Hierarchical Framework for Constructing Intelligent Systems Metrics

II Day, Morning B: Modeling and Measuring Machine Intelligence
Co-Chairs: P. Davis, T. Whalen

- P. Davis, Exploratory Analysis Enabled by Multiresolution, Multiperspective Modeling
- M. Jabri, Measuring intelligence: a neuromorphic perspective
- I. Nourbakhsh , Two measures for measuring the 'intelligence' of human-interactive robots in contests and in the real world: perceptiveness and expressiveness
- T. Whalen, What is the Value of Intelligence and How Can It Be Measured?

II Day, Morning C: Evaluating Factors of Intelligence in Systems
Co-Chairs: J. Hernandes-Orallo, C. Peterson

- J. Hernandes-Orallo, On the Computational Measurement of Intelligence Factors
- A. Wild, Heterogeneous Computing
- J. Bryson, et al, Hypothesis Testing for Complex Agents
- T. Balch, Hierarchic Social Entropy: An Information Theoretic Measure of Robot Group Diversity

Lunch 12.30 PM — 2 PM

PLENARY LECTURE — 2 PM — 3 PM

**S. Grossberg, Some Constraints on Intelligent Systems:
Autonomous Computation in a Changing World**

Coffee Break — 3 PM — 3.15 PM

Sessions: 3.15 PM — 5.15 PM

II Day, Afternoon A: Measuring of Intelligence of Multiagent Networks

Chair and Organizer: S. Phoha

- R. R. Brooks, STIGMERGY: A measure of intelligence for emergent distributed behaviors
- S. Phoha, D. Friedlander, Goodness of Fit Measures for Intelligent Behaviors of Interacting Machines
- M. E. Cleary, M. Abramson, M. B. Adams, S. Kolitz. Metrics for Embedded Collaborative Intelligent Systems
- D. Friedlander, S. Phoha, A. Ray, Domain Independent Measures of Intelligent Control
- S. Perraju Tolety, G. Uma, On Measuring Intelligence in Multi-Agent Systems

II Day, Afternoon B: Evaluating Intelligent Systems by Testing and Competition: Benchmarks

Co-Chairs and Organizers: A. Schultz, R. Murphy

- A. Schultz, Evolution of Metrics for Mobile Robots
- A. Jacoff, E. Messina, J. Evans, A Standard Test Course for Urban Search and Rescue Robots
- R. Murphy, J. Casper, M. Micire, J. Hyams, Assessment of the NIST Standard Test Bed for Urban Search and Rescue Competitions
- T. Balch, Performance/N is the Wrong Metric for Multirobot Teams
- S. K. Agrawal, A. M. Ferreira, S. Pledgie, Performance Evaluation of Robotic Systems: A Proposal for a Benchmark problem

II Day, Afternoon C: Measuring Intelligence of Distributed Systems

Co-Chairs: R. Fakory, W. J. Davis

- W. J. Davis, Evaluating Performance of Distributed Intelligent Control System
- R. Fakory, M. Jahangiri, Real Time Distributed Expert System for Automated Monitoring of Key Monitors in Hubble Space Telescope
- X. Qin, A. E. Aktan, Distributed Internet-Based Multi-Agent Intelligent Infrastructure System
- D. P. Gravel, W. S. Newman, Flexible Robotic Assembly

Plenary Discussion- 5.15 PM — 6.15 PM

Panel: T. Balch, P. Davis, W. J. Davis, R. Fakory, J. Hernandes-Orallo, R. Murphy, S. Phoha, T. Samad, A. Sanderson, A. Schultz, T. Whalen

Evening: COCKTAILS AND BANQUET
— 6.45 PM at "Holiday Inn"

L. Zadeh,

Banquet speech "The Search for Metrics of Intelligence -- A Critical View."

3rd Day, Wednesday, August the 16th

PLENARY LECTURE — 9 AM — 10 AM

W. Freeman, The neurodynamics of intentionality in animal brains provides a basis for constructing devices that are capable of intelligent behavior

Coffee Break: 10 AM-10.30 AM

Sessions: 10.30 AM — 12.30 PM

III Day, Morning A: Measuring Intelligence Taking in Account Linguistic, Psychological and Biological Factors
Co-Chairs: L. Reeker, A. Meystel

- L. Reeker, Theoretical Constructs for Measurement Performance and Intelligence
- A. Meystel, Generalizing Natural Language Representations for Measuring the Intelligence of Systems
- P. Wang, Machine Intelligence Ranking
- A. Treister-Goren, J. Dunietz, The AI Language Development Metric

III Day, Morning B: Measuring Intelligence of Systems with Autonomy and Mobility
Co-Chairs: G. S. Sukhatme, J. Weng

- G. S. Sukhatme, Measuring Mobile Robots Performance: Approaches and Pitfalls
- L. E. Parker, Evaluating Success in Autonomous Multi-robot Teams: Experience of ALLIANCE Architectures Implementation
- A. Lacaze, S. Balakirsky, Search Graph Formation for Minimizing the Complexity of Planning
- J. Weng, Automatic Mental Development and Performance Metrics for Intelligent Systems

Lunch 12.30 PM — 2 PM

PLENARY LECTURE — 2 PM — 3 PM

A. Meystel , Evolution of Intelligent Systems Architectures: What Should Be Measured

Coffee Break — 3 PM — 3.15 PM

Afternoon Session — 3.15 PM — 5.15 PM

III Day, Afternoon (Plenary Panel): Perspectives of Governmental Programs on Measuring Intelligence
Panel organizers — J. Albus, JBlitch, J. Evans

- J. Albus, NIST
- J. Blitch, DARPA
- J. Evans, NIST
- C. Shoemaker ARL,
- C. Weisbin, NASA

General Discussion of the Workshop Results- 5.15 PM — 6.15 PM

Panel: J. Albus, J. Evans, E. Messina, A. Meystel, L. Reeker, G. S. Sukhatme, J. Weng

The Meeting is adjourned 6.15 PM

APPENDIX C
AUTHOR INDEX

☆U.S. GOVERNMENT PRINTING OFFICE:2001-472-068/55651

NIST Technical Publications

Periodical

Journal of Research of the National Institute of Standards and Technology—Reports NIST research and development in those disciplines of the physical and engineering sciences in which the Institute is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Institute's technical and scientific programs. Issued six times a year.

Nonperiodicals

Monographs—Major contributions to the technical literature on various subjects related to the Institute's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NIST, NIST annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NIST under the authority of the National Standard Data Act (Public Law 90-396). NOTE: The Journal of Physical and Chemical Reference Data (JPCRD) is published bimonthly for NIST by the American Institute of Physics (AIP). Subscription orders and renewals are available from AIP, P.O. Box 503284, St. Louis, MO 63150-3284.

Building Science Series—Disseminates technical information developed at the Institute on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NIST under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NIST administers this program in support of the efforts of private-sector standardizing organizations.

Order the **following** *NIST publications—FIPS and NISTIRs—from the National Technical Information Service, Springfield, VA 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NIST pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NIST Interagency or Internal Reports (NISTIR)—The series includes interim or final reports on work performed by NIST for outside sponsors (both government and nongovernment). In general, initial distribution is handled by the sponsor; public distribution is handled by sales through the National Technical Information Service, Springfield, VA 22161, in hard copy, electronic media, or microfiche form. NISTIR's may also report results of NIST projects of transitory or limited interest, including those that will be published subsequently in more comprehensive form.

U.S. Department of Commerce
National Institute of Standards
and Technology
Gaithersburg, MD 20899–0001

Official Business
Penalty for Private Use $300